

EpiData Software for Operations Research in Tuberculosis Control

A course developed by EpiData Promoters* in collaboration with the EpiData Association and the International Union Against Tuberculosis and Lung Disease



*** EpiData Promoters:**

Hans L **Rieder**, Kirchlintach, Switzerland

Chiang Chen-Yuan, Taipei, Taiwan

Sven Gudmund **Hinderaker**, Bergen, Norway

Achilles **Katamba**, Kampala, Uganda

Ajay M V **Kumar**, Bangalore, India

Nguyen Binh **Hoa**, Hanoi, Vietnam

Marcel **Zwahlen**, Berne, Switzerland

Zaw Myo **Tun**, Mandalay, Myanmar

Hans L Rieder for the EpiData Promoters

Date last revised: April 9, 2012

Note on versions of EpiData software: Always use the most recent release version of EpiData software which can be obtained freely from the EpiData website <http://www.epidata.dk>. The course is updated at least whenever a new version requires adaptation or following an in-class course, whichever comes earlier.

Course content

Part A: EpiData Entry

- Exercise 1 A simple questionnaire
- Exercise 2 The QES-REC-CHK triplet
- Exercise 3 Using Labelblocks instead of legal values
- Exercise 4 Adding field labels and value labels
- Exercise 5 Data entry and validation
- Exercise 6 Data safety
- Exercise 7 Using an external file for Labelblocks
- Exercise 8 Dealing with incomplete dates
- Exercise 9 Keeping track of data entry time

Part B: EpiData Analysis

- Exercise 1 A basic program
- Exercise 2 Appending and making new REC files
- Exercise 3 Creating a string variable
- Exercise 4 Aggregating data

Part C: Operations research

- Exercise 1: Creating a working dataset
- Exercise 2: Variability in serial smears
- Exercise 3: Incremental yield from serial smears
- Exercise 4: Confirmatory results in serial smears

Part D: More on EpiData software

- Exercise 1: Relational database and aggregating vs from “Long-to-wide”
- Exercise 2: A statistical process control chart
- Exercise 3: A simplified survival analysis
- Exercise 4: Creating a menu for standard reports

Introduction

In the 1980ies, the United States Centers for Disease Control and Prevention (CDC) developed public health software for its Epidemic Intelligence Service. The software package was called Epi Info and experienced an unprecedented spread throughout the world's public health community and was actively supported by the World Health Organization.

The initiative to make EpiData was taken by Jens M Lauritsen from Denmark. Initially it was conceived of as part of the "Initiative for Accident Prevention" at Funen County. Why was it necessary to develop a new data entry program, if Epi Info version 6 had all that was needed in terms of control of data entry and simplicity? With the development of the Windows[®] operating system the majority of current computer users found it increasingly harder to cope with the DOS[®] operating system of working in Epi Info. Furthermore, with the change in the operating system of computers to newer versions of the Windows[®] operating system, the Epi Info 6.04d based on the DOS[®] operating system is eventually becoming obsolete. Nevertheless, the principles underlying Epi Info were fundamentally sound and needed to be preserved for the new generation of public health practitioners grown up with the Windows[®] operating system.

On the Epi Info discussion list there were some discussions on strategies around 1997-1998, when the Epi Info team at CDC in USA decided to make an updated Epi Info version 2000. The updated Epi Info applies a different strategy in using a completely new way of working and basing it on the Microsoft Access[®] database format instead of simple text files (ASCII).

Commercially available proprietary software is not focused on documentation, simplicity of use and validation of double-entered data. EpiData Entry is a program with a focus on data entry. The ambition of EpiData Entry was to create a simple to use independent application, which would not interfere with or require any special database system drivers (dll-based routines) shared with or interfering with other applications.

EpiData software consists now of two modules, EpiData Entry and EpiData Analysis. In this course you will be learning the use of both EpiData Entry (Part A) and EpiData Analysis (Part B), and to apply them to operations research (Part C), starting with its very basic functionality to increasing sophistication, and finally extend your programming skills in EpiData Analysis (Part D).

Course Objective:

Acquire the skills to capture research study data of high quality, supported by thorough documentation of every procedure, conduct basic analyses, and apply them to operations research

History of the course

The data for the original course were collected in a study simultaneously conducted in Benin, Malawi, Nicaragua, and Senegal.

The persons who collaborated in study design, data collection, analysis, and writing of a report were Séverin **Anagonou** (Benin), Thuridur **Arnadottir** (The Union), Fatoumata **Ba** (Senegal), Awa Héléne **Diop** (Senegal), Donald A **Enarson** (The Union), Martin **Gninafon** (Benin), A C **Kasalika** (Malawi), Hans L **Rieder** (The Union), Tone **Ringdal** (The Union), Felix L M **Salaniponi** (Malawi), Alejandro A **Tardencilla Gutierrez** (Nicaragua) and Arnaud **Trébucq** (The Union).

Utilizing the data generated in this collaborative research project, **three courses** of two weeks duration each **from 1997 to 1999** with a total of 18 participants were conducted in Paris by The Union. Subsidies to defray costs of these three courses were borne by the United States Centers for Disease Control and Prevention, the Coopération Française, the Norwegian Heart and Lung Association, the International Organization for Migration, and the Korean Institute of Tuberculosis.

The **first course** was held in **April 1997** in Paris, France. The participants were Francis **Adatu-Engwau** (Uganda), Nora **Bonso-Bruce** (Ghana), Awa Héléne **Diop** (Senegal), Asma **Elsony** (Sudan), and Amina **Jindani** (The Union). The facilitators were Thuridur **Arnadottir**, Eric **Brenner** (USA), Lawrence J **Geiter** (The Union), Hans L **Rieder** (The Union), and Arnaud **Trébucq** (The Union).

The **second course** was held in **April 1998** in Paris, France. The participants were Mohammed **Akhtar** (Pakistan), Maurice **Andriamiandrisoa** (Madagascar), Manfred **Danilovits** (Estonia), Lew Woo Jin (Korea, Republic of), **Nguyen** Phuong Hoa (Vietnam), Shanta Bahadur **Pande** (Nepal), and Alejandro A **Tardencilla Gutierrez** (Nicaragua). The facilitators were Thuridur **Arnadottir** (The Union), Eric **Brenner** (USA), Lawrence J **Geiter** (The Union), Hans L **Rieder** (The Union), and Arnaud **Trébucq** (The Union).

The **third course** was held in **April 1999** in Paris, France. The participants were Ademir **de Albuquerque Gomes** (Brazil), Fatoumata **Ba** (Senegal), Ferdinand **Kassa** (Benin), Pushpa **Malla** (Nepal), **Tieng** Sivanna (Cambodia), and Wang Jie-Siu (China), and. The facilitators were Thuridur **Arnadottir** (The Union), Eric **Brenner** (USA), Lawrence J **Geiter** (The Union), Hans L **Rieder** (The Union), and Arnaud **Trébucq** (The Union).

Because of the large costs associated with conducting the course, the course content and format was tested through interactive distance learning during the year 2000 by email. We are particularly indebted to Panganai **Dhliwayo** (Zimbabwe) and Robert **Makombe** (Zimbabwe) who went through this course in their free time. This helped in clarifying ambiguities and removing errors and uploading the course in September 2000 to the Internet (<http://www.tbrieder.org>).

The **fourth course** was held in Addis Ababa, Ethiopia, in **April 2001**. The participants were Ahmed **Abdurehman** (Ethiopia), Jemal **Aliy** (Ethiopia), Mekdes Gebeyehu **Ayicheh** (Ethiopia), Abebe **Habte** (Ethiopia), Yohannes **Mengistu** (Ethiopia), Moustapha **Ndir** (Senegal), Serkalem **Tadesse** (Ethiopia), Betru **Tekle** (Ethiopia), Mohammed Ahmed **Yassin** (Ethiopia), and Getachew Eyob **Yitelelu** (Ethiopia). The facilitators were Nils E **Billo** (The Union), Panganai **Dhliwayo** (Zimbabwe), and Hans L **Rieder** (The Union).

In 2002, the course material was once more revised to replace Epi Info 6 with EpiData Entry for questionnaire design, data checks, data entry, and data validation. The analysis of the data was continued to be done with the 6.04d DOS version of Epi Info.

The **fifth course** was held in Paris, France, in **January 2003**. The participants were Nadia **Aït-Khaled** (The Union), Kya Jai Maug **Aung** (Bangladesh), **Chiang** Chen-Yuan (Taiwan), Paula I **Fujiwara** (The Union), Achilles **Katamba** (Uganda), **Kim** HeeJin (Korea, Republic of), Dumitru **Laticevschi** (Moldova), Jones **Michongwe** (Malawi) and Jotam G **Pasipanodya** (Zimbabwe). The facilitators were Panganai **Dhliwayo** (Zimbabwe), Robert **Makombe** (Zimbabwe), and Hans L **Rieder** (The Union).

The **sixth course** was held in Yangon, Myanmar, in **December 2003**. The participants were Nyein Nyein **Aye** (Myanmar), May Yee **Chan** (Myanmar), Tin Mi Mi **Khaing** (Myanmar), Thandar **Lwin** (Myanmar), Htar Htar **Oo** (Myanmar), Saw **Thein** (Myanmar), Ti **Ti** (Myanmar), Myo **Zaw** (Myanmar), and Thin Thin **Yee** (Myanmar). The facilitators were **Chiang** Chen-Yuan (The Union), Hans L **Rieder** (The Union), and Ira D **Rusen** (Canada). In all previous courses the hypothesis was to test whether The Union's assumption that ten suspects needed to be examined to identify one case of tuberculosis was applicable to the study area. During the course it emerged, however, that even refutation of the hypothesis had relatively little programmatic implication. It was thus decided to integrate the previously supplementary exercises (dealing with variability of positive findings, patterns of recorded results, and potential incremental yield) into one single hypothesis addressing the critical value of the number of smear examinations that may be justified to identify one additional case or failure on the third diagnostic or the second follow-up smear examination, respectively.

The **seventh course** was held in Paris, France, in **January 2004**. The participants were Wafaa Hassan **Ali Taha** (Sudan), Goar **Balasanants** (Russia), Nulda **Beyers** (South Africa), Kathy **Lawrence** (South Africa), Biggie **Mabaera** (Zimbabwe), Nymadawaa **Naranbat** (Mongolia), Mahshid **Nasehi** (Iran), Mauro **Occhi** (Mozambique), Yevgeniy **Shubin** (Russia), and Abigail **Wright** (World Health Organization). The facilitators were Panganai **Dhliwayo** (Zimbabwe), Jens M **Lauritsen** (EpiData Association, Denmark), Robert **Makombe** (Zimbabwe), and Hans L **Rieder** (The Union).

The **eighth course** was held in Berne, Switzerland, in **July 2004**. The participants were Sondhja **Bitter** (Switzerland), Eva **Blozik** (Switzerland), Lorenz **Borer** (Switzerland), Michael **Endrich** (Switzerland), Karin **Imoberdorf-Baumgartner** (Switzerland), Caroline **Keller** (Switzerland), Hansjörg **Lüthy** (Switzerland), Christoph **Pammer** (Austria), Sabine **Recker** (Switzerland), Kurt **Schmidlin** (Switzerland), Jan **Wagner** (Switzerland), Sabine **Walser** (Austria), and Mark **Witschi** (Benin). The facilitators were Hans L **Rieder** (The Union) and Marcel **Zwahlen** (University of Berne, Switzerland). The curriculum of this course was changed in two important aspects from the previous courses. First, a core curriculum was developed to allow the conduct of the course within five working days without loss of any relevant course aspects. Refinements of specific aspects were offered in addenda independent of each other. This allowed course completion within one week (core curriculum) or eight days (core and extended curriculum). Second, the Epi Info Analysis component was replaced by the trial and testing version of EpiData Analysis which proved to be working smoothly without any glitches.

The **ninth course** was held in Paris, France, in **January 2005**. The participants were Anneke **Hesseling** (South Africa), **Hu** Dongmei (China), Zanele **Hwalima** (Zimbabwe), **Liu** Zhentian (China), Henri **Luwaga** (Uganda), **Nguyen** Thien Huong (Vietnam), John **Osho** (Nigeria), **Tran** Ngoc Buu (Vietnam), Nevin **Wilson** (The Union), and **Yao** Hongyan (China). The facilitators were Panganai **Dhliwayo** (The Union), Jens M **Lauritsen** (EpiData Association, Denmark), Robert **Makombe** (Zimbabwe), and Hans L **Rieder** (The Union). In this course, only free public domain software was used. Data management and analysis were done with

EpiData and EpiData Analysis, supplemented where needed with spreadsheet functions of OpenOffice.

The **tenth course** was held in Berne, Switzerland, in **July 2005**. The participants were Thomas **Bart** (Switzerland), Lisanne **Christen** (Switzerland), Stephanie **Christensen** (Switzerland), Michael **Flück** (Switzerland), Yvonne **Jansen** (Switzerland), Irène **Marty** (Switzerland), Jeanne **Moser** (Switzerland), Christoph **Napierala** (Switzerland), Barbara **Prokup** (Germany), Martin **Raab** (Switzerland), Franziska **Rabenschlag-Trösch** (Switzerland), Claudia **Sauerborn** (Switzerland), and Yasemin **Yüksel** (Switzerland). The facilitators were Hans L **Rieder** (The Union) and Marcel **Zwahlen** (University of Berne, Switzerland). This course used the same software as the January 2005 course with the latest pre-release version of EpiData Analysis (Version 0.9, Release 5, Build 26). Input from the participants and faculty led to further streamlining of some exercises.

The **eleventh course** was conducted in Paris, in **January 2006**. The participants were **Chay Sokun** (Cambodia), Andrew D R C **Dimba** (Malawi), Elhafiz **Hussein Ibrahim** (Sudan), Akramul **Islam** (Bangladesh), Suksont **Jittimane** (Thailand), **Le Van Duc** (Vietnam), **Nguyen Binh Hoa** (Vietnam), Helmuth **Reuter** (South Africa), Ezra **Shimeles** (The Union), and **Xu Min** (China). The facilitators were Achilles **Katamba** (Uganda), Jens M **Lauritsen** (EpiData Association, Denmark), and Hans L **Rieder** (The Union). In September 2005, the EpiData Association released the stable EpiData Analysis Version 1.0. Between release and course commencement, the EpiData Association released upgrade Version 1.1 which added the last functionality (aggregate command) that had been possible previously only in Epi Info DOS Version 6.

During 2005 and 2006 the structure of the course was changed into three parts (EpiData Entry, EpiData Analysis, and Operations Research.). Parts A (EpiData Entry) and B (EpiData Analysis) were stripped of the operations research component for the benefit of those who do not have sufficient time available or who wish to familiarize themselves solely with the software. Conversely, Part C (Operations research) was stripped of components now introduced in Parts A and B, concentrating on the application of the latter to the example research project.

In July 2006, the **twelfth course** was conducted in Berne, Switzerland, for Master of Public Health students and other interested participants. These were Martin **Adam** (Switzerland), Edith **Betschart** (Switzerland), Tobias **Eckert** (Switzerland), Denise **Felber Dietrich** (Switzerland), T John **Kessler-Teuscher** (Switzerland), Esther **Kolb** (Switzerland), Brigitte **Kuenzle** (Switzerland), Sonia **Menéndez-González** (Switzerland), Stefan **Neuner-Jehle** (Switzerland), Christoph Paul **Röder** (Switzerland), and Hildebrand **Schwab** (Switzerland). The facilitators were Hans L **Rieder** (The Union) and Marcel **Zwahlen** (University of Berne, Switzerland).

In August 2006, the **thirteenth course** was given in Changsha, Hunan Province, China. All participants were from China. They were 范月玲 (**Fan Yueling**), 贾忠彬 (**Jia Zhongbin**), 阚晓宏 (**Kan Xiaohong**), 李晓凤 (**Li Xiaofeng**), 林岩 (**Lin Yan**), 邱柏红 (**Qiu Baihong**), 谭振 (**Tan Zhen**), 王铂 (**Wang Bo**), 汪清雅 (**Wang Qingya**), 张恩溥 (**Zhang Enpu**), 张宏伟 (**Zhang Hongwei**), 张会民 (**Zhang Huimin**), and 赵丁源 (**Zhao Dingyuan**). Facilitators were **Chiang Chen-Yuan** (The Union) and Hans L **Rieder** (The Union).

In December 2006, the **fourteenth** and **fifteenth courses** were given sequentially in Beijing, China. All participants were from China. In the thirteenth course the participants were 陈慧娟 (**Chen Huijuan**), 陈静 (**Chen Jing**), 陈伟 (**Chen Wei**), 房宏霞 (**Fang Hongxia**), 胡嘉 (**Hu Jia**), 梁路 (**Liang Lu**), 刘二勇 (**Liu Eryong**), 马斌忠 (**Ma Binzhong**), 王丹霞 (**Wang Danxia**), 吴顶峰 (**Wu**

Dingfeng), 于宝柱 (**Yu Baozhu**), and 张慧 (**Zhang Hui**). In the fourteenth course, the participants were 陈广华 (**Chen Guanghua**), 孔霞 (**Kong Xia**), 李曙光 (**Li Shuguang**), 罗丹 (**Luo Dan**), 马艳 (**Ma Yan**), 庞学文 (**Pang Xuewen**), 司马雅云 (**Sima Yayun**), 徐吉英 (**Xu Jiyin**), 依帕尔 (**Yi Pa'er**), 张广恩 (**Zhang Guang'en**), 赵津 (**Zhao Jin**), and 周扬 (**Zhou Yang**). Facilitators were Jens M **Lauritsen** (EpiData Association, Denmark), Biggie **Mabaera** (University of Zimbabwe, Zimbabwe), and Hans L **Rieder** (The Union), with the assistance of 胡冬梅 (**Hu Dongmei**), 徐敏 (**Xu Min**), and 张慧 (**Zhang Hui**).

In June 2007, the **sixteenth course** was given in Khartoum, Sudan. Two major changes were made to the course curriculum. The first change was that the database for Part C was changed to the utilization of the dataset collected as a course project of the January 2003 and 2004 courses (data courtesy: Dumitru **Laticevschi**, Moldova; Nymadawaa **Naranbat**, Mongolia; Achilles **Katamba**, Uganda; Biggie **Mabaera**, Zimbabwe). The second change was to use the test version 2.0 of EpiData Analysis. All participants were from Sudan. The participants were خديجة ادم محمد (Khadiga **Adam** Mohammed), اسرار محمد عبدالسلام (Asrar M A/Salam **Elegail**), معاذ سرالختم اسماعيل (Maaz Sier Elkhatim Ismail), حباب خالد الخير (Habab Khalid **Elkheir** Omer), امل السمانى اسماعيل (Amel Elsammani Mohamed Ahmed **Elmuozamil**), مناضل حسن محمد علي (Monadil **Hassan** Mohamed Ali), سيد محمد همت (Sayed Mohammed Shareef **Himat**), ثويبة عمر علي محقر (Thoeiba Omer Ali **Muhagger**), عبدالمجيد عثمان موسى (Abdelmageed Osman **Musa**), علا محمود رحمة الله (Olla Mahmoud **Rahamtalla**) and عزمي عبدالرحمن عمارة (Azmi **Omara**). The coordinator was لوران علي زين العابدين (Louran **Zein Abdin Ali**, National Tuberculosis Programme Sudan), and facilitators were الحافظ حسين ابراهيم (Elhafiz Hussein **Ibrahim**, Epi-Lab, Sudan), and Hans L Rieder (The Union).

In July 2007, the **seventeenth course** was given in Berne, Switzerland, for Master of Public Health students and other interested participants. These were Nicole **Bender-Oser** (Switzerland), Bettina **Bringolf-Isler** (Switzerland), Sabina **Büttner** (Switzerland), Christian **Frei** (Switzerland), Florian **Gutzwiller** (Switzerland), Peter **Heri** (Switzerland), Kerstin **Hug** (Switzerland), André B **Kind** (Switzerland), Dimitri **Korol** (Switzerland), Cornelia **Marti** (Switzerland), Anne **Spaar** (Switzerland), and Françoise **Teuscher** (Switzerland). The facilitators were Hans L **Rieder** (The Union) and Marcel **Zwahlen** (University of Berne, Switzerland).

In November 2007, the course material was updated to make minor changes to Part A and to accommodate the changes introduced with the release Version 2.0 of EpiData Analysis.

In April 2008, the **eighteenth course** was given in Chisinau, Moldova. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.0 (Pre-release 2.1 Test Build 132) was utilized. The participants were Əlixanova Natəvan (**Alikhanova** Natavan), Azerbaijan; uCa nanava (Ucha **Nanava**), Georgia; maia qavTaraZe (Maia **Kavtaradze**), Georgia; Otilia **Scutelnicu**, Moldova; Angela **Capcelea**, Moldova; Rita **Seicas**, Moldova; Viorel **Soltan** Moldova; Ştefan **Savin**, Moldova; nino lomTaZe (Nino **Lomtadze**), Georgia; Constantin Dan **Nicolaiciu**, Romania; Iuliana **Husar**, Romania; Richard **Oleko** (Sudan). Facilitators were Jens M **Lauritsen** (EpiData Association, Denmark), Biggie **Mabaera** (Temporary Consultant to The Union, Lesotho), and Hans L **Rieder** (The Union).

In June 2008, the **nineteenth course** was given in Ulaanbaatar, Mongolia. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.0 (Pre-release 2.1 Test Build 132) were utilized. The participants were Пүрэвдагва Анузаяа (**Anuzaya** Purevdagva), Цолмон Билэгтсайхан (**Bilegtsaikhan** Tsolmon), Бүрнээбаатар Буюнхишиг (Burneebaatar **Buyankhishig**), Бүдбазар Энхтуяа (Budbazar **Enkhtuya**), Довдон Хандаасүрэн (Dovdon **Khandaasuren**), N **Naranbold** (Н. Наранболд), Ваатархуу Оюунтуяа (Баатархүү Оюунтуяа), Dorj **Otgontsetseg** (Дорж Отгонцэцэг), Demberelsuren **Sodbayar**

(Дэмбэрэлсүрэн **Содбаяр**), Sandagdorj **Tuvshingerel** (Сандагдорж **Түвшингэрэл**), Luvsansharav **Ulzii-Orshikh** (Лувсаншарав **Өлзий-Орших**). Hans L Rieder (The Union) was the facilitator. Following are the names of Nymadawa **Naranbat**, who organized the course, and the eleven participants written in the traditional Mongolian alphabet.



In June / July 2008, the **twentieth** course was given in Berne, Switzerland, for Master of Public Health students and other interested participants. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.0 (Pre-release 2.1 Test Build 132) were utilized. All participants were from Switzerland. They were Caroline **Bähler-Baumgartner**, Daniela **Dyntar**, Martin **egger**, Carola A **Huber**, Ursula **Kälin-Keller**, Annette **Koller Doser**, Andrea **Merkel-Hoek**, Eric **Odenheimer**, Helen **Prytherch**, Anna **Späth**, and Melinda **Spiesshofer**). The facilitators were Hans L **Rieder** (The Union) and Marcel **Zwahlen** (University of Berne, Switzerland).

In October 2008, the **twenty-first** course was given in Kampala, Uganda. Course I was given during this five-day course. It was specifically designed to fit this 5-day course, also incorporating the previously existing brief overview of EpiData software. EpiData Entry Version 3.1 (Build 270108) and EpiData Analysis Version 2.0 (Pre-release 2.1 Test Build 141) were utilized. The participants were Raymond **Asiimwe** (Uganda), Bernard Ssentalo **Bagaya** (Uganda), Freddy **Bwanga** (Uganda), Henry **Byabajungu** (Uganda), Basra Esmail **Doulla** (Tanzania), Nicholas **Ezati** (Uganda), Fred **Kangave** (Uganda), George **Lukyamuzi** (Uganda), Diana **Nagunga** (Uganda), and Raymond **Shirima** (Tanzania). Francis **Adatu-Engwau** (Uganda) was a guest participant, and formerly a participant in the first course. The facilitator was Hans L **Rieder** (The Union), supported in part by Achilles **Katamba** (Makerere University, Uganda, and The Union Country Office, Uganda).

In February 2009, the **twenty-second** course was given in Kampala, Uganda. Course II, Part B (EpiData Analysis) and Part C (Operations research) were given during this five-day course, although some exercises had to be skipped due to the brief duration of the course. EpiData Analysis Version 2.1 (Test Build 159) was utilized. The participants were Francis **Adatu-Engwau** (Uganda), Bernard Ssentalo **Bagaya** (Uganda), Freddy **Bwanga** (Uganda), Basra Esmail **Doulla** (Tanzania), Fred **Kangave** (Uganda), Diana **Nagunga** (Uganda), and Raymond **Shirima** (Tanzania). The facilitator was Hans L **Rieder** (The Union), supported in part by Achilles **Katamba** (Makerere University, Uganda, and The Union Country Office, Uganda).

In July 2009, the **twenty-third** course was given in Berne, Switzerland, for Master of Public Health students and other interested participants. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.2 (Build 169) were utilized. All but one participant from the principality of Liechtenstein were residents of Switzerland. They were Aline **Barbir**, Rita

Born, Mazda Farshad, Oliver Fuchs, Juliette Gerber, Gerti Kitting Gaillard, Meltem Kutlar Joss, Teresa Leisebach Minder, Elisabeth Oberfeld, Anna Plym, Corinna Rüegg, Dino Schlamp, Eliane Siegenthaler, Federico Soldati, and Esther Walser (Principality of Liechtenstein). The facilitators were Hans L **Rieder** (The Union) and Marcel **Zwahlen** (University of Berne, Switzerland).

In October 2009, the **twenty-fourth** course was given in Paris, France, for fellows and participants in the training module 2 offered by the Centre for Operational Research of The Union. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.2.1.171 were utilized. The participants were Dawit **Assefa Lemma** (Ethiopia), Walter Kizito **Kibango** (Kenya), Proscovia Namuwenge **Mukonzo** (Uganda), Mweete Debra **Nglazi** (South Africa), **Nguyen Binh Hoa** (Viet Nam), Mahfuza **Rifat** (Bangladesh), Srinath **Satyanarayana** (India), Zaw Myo **Tun** (Myanmar), Hannock Mukoma **Tweya** (Malawi) and Susanna Sophia **Van Wyk** (South Africa). The facilitators were Hans L Rieder (The Union) and Anthony D Harries (The Union), assisted by **Nguyen Binh Hoa** (Fellow, Viet Nam).

In March 2010, the **twenty-fifth** course was given in Paris, France, for three selected fellows and participants in the training module 2 offered by the Centre for Operational Research of The Union. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.2.1.171 were utilized. The course focused on Parts C (Operations Research) and D (More on EpiData software). The participants **Nguyen Binh Hoa** (Viet Nam), Zaw Myo **Tun** (Myanmar), and Hannock Mukoma **Tweya** (Malawi). The facilitators were Hans L **Rieder** (The Union) and Jens M **Laurtisen** (EpiData Association).

In July 2010, the **twenty-sixth** course was given in Berne, Switzerland, for Master of Public Health students and other interested participants. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.2 (Build 171) were utilized. All participants were residents of Switzerland. They were Brigitte **Brunner**, Adrian **Businger**, Elisabeth **Maurer Schild**, Rebecca **Osterwalder**, Alexandra **Rauch**, Charles **Senessie**, Ulf **Tölle**, and Roco **Umbescheidt**. The facilitators were Hans L **Rieder** (The Union) and Marcel **Zwahlen** (University of Berne, Switzerland).

In October 2010, the **twenty-seventh** course was given in Paris, France, for fellows and participants in the training module 2 offered by the Centre for Operational Research of The Union. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.2.1.171 were utilized. The participants were Felix **Afutu** (Ghana), Sarabjit Singh **Chadha** (India), Karen **Du Preez** (South Africa), Razia **Fatima** (Pakistan), Oliver **Gadabu** (Malawi), Lucy **Guluka-Gawa** (Malawi), Mohammed **Kogali** (Sudan), Rose Jepchumba **Kosgei** (Kenya), Ajay Kumar **Madhugiri Venkatachalaiah** (India), Tonderayi Clive **Murimwa** (Zimbabwe), Mauro **Niskier Sanchez** (Brazil), and Kudakwashe Collin **Takarinda** (Zimbabwe). The facilitators were Hans L **Rieder** (The Union), **Nguyen Binh Hoa** (Viet Nam), Zaw Myo **Tun** (Myanmar), and Sven Gudmund **Hinderaker** (The Union).

In May 2011, the **twenty-eighth** course was given in Paris, France, for two selected fellows from the training module 2 offered by the Centre for Operational Research of The Union. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.2.1.171 were utilized. The course focused on Parts C (Operations Research) and D (More on EpiData software). The participants Karen **Du Preez** (South Africa) and Ajay **Kumar Madhugiri Venkatachalaiah** (India). The facilitators were Hans L **Rieder** (The Union), Jens M **Laurtisen** (EpiData Association), and Hannock Mukoma **Tweya** (Malawi).

In November 2011, the twenty-ninth course was given in Paris, France for fellows and participants in the training module 2 offered by the Centre for Operational Research of The

Union. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.2.1.171 were utilized. The participants were Henry Shadreck **Kanyerere** (Malawi), Nicholas **Kirui** (Kenya), Pranay **Lal** (India), Sharath Bugurina **Nagaraja** (India), Sharan **Ram** (Fiji), Carlos Alberto **Mendoza-Ticona** (Peru), Emmanuel **Singogo** (Malawi), Nelda **van Soelen** (South Africa), Kerry **Viney** (New Caledonia), and Aung Naing **Win** (Myanmar). The facilitators were Hans L **Rieder** (The Union), Sarabjit **Chadha** (The Union), and Ajay **Kumar** (India).

In February 2012, the **thirtieth** course was given in Kathmandu, Nepal for fellows and participants in the training module 1b offered by the Centre for Operational Research of South East Asia office of The Union. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.2.1.171 were utilized. The participants were Tshering **Jamtsho** (Bhutan), Tashi **Denup** (Bhutan), Shyla **Islam** (Bangladesh), Wasantha **Jayakodi** (Sri Lanka), Swati **Srivastava** (India), Suresh **Shastri** (India), Ramya **Ananthakrishna** (India), Rajendra **Basnet** (Nepal), Caetano **Gusmao** (Timor Leste), Sokhan **Khann** (Cambodia), Sarwat **Shah** (Pakistan), Iwayan **Artwan** (Indonesia). The facilitators were Ajay M V **Kumar** (The Union, India) and Srinath **Satyanarayana** (The Union, India).

In April 2012, the **thirty-first** course was given in Paris, France, for three selected fellows from the training module 2 offered by the Centre for Operational Research of The Union. EpiData Version 3.1 (Build 270108) and EpiData Analysis Version 2.2.1.178 were utilized. The course focused on Parts C (Operations Research) and D (More on EpiData software). The participants were Sarabjit Singh **Chadha** (India), Sharath **Burugina Nagaraja** (India), and Nicholas **Kirui** (Kenya). The facilitators were Hans L **Rieder** (The Union), Ajay M V **Kumar** (The Union, India), and Jens M **Laurtisen** (EpiData Association).

Preparatory steps before you begin

We will be using solely legal and free software, which you are free to distribute further. The software we are going to use is:

Acrobat Reader

EpiData

EpiData Analysis

Installation of Acrobat Reader™

All module texts are *.PDF (**P**ortable **D**ocument **F**ormat) files which you can only read with the freely available Acrobat Reader™. If you have an Internet connection, you should visit the website of Adobe company:

www.adobe.com

to obtain the most recent version. If you do not have such a connection, you will find the software in the “Software” section of this website / CD-ROM.

Click on “Acrobat Reader” in the “Software” section, save it (not open) it to a directory of your choice. From within Windows Explorer, double-click the downloaded file, and follow the instructions.

If the course comes on a CD-ROM

If your course is on a CD-ROM, it might be easiest to copy its entire content, i.e., the folder containing the course, to your computer hard disk. One level below the folder, you will see a file:

index.html

It is a commonly used convention that the opening page (the “home page”) of a Web site is given this name. The course material is organized like a Web site. If you double-click on this file name (INDEX.HTML), the Web opens. For fastest access to the Web you might wish to make a short-cut to this opening page on your desktop and then drag it down to quick launch task bar. This will always give you visible access with a single click.

Installation of EpiData Entry and EpiData Analysis

EpiData Entry is a very small executable file. Although it is Windows-based, it does not interfere with your Windows set-up: all files are placed in a single directory without leaving any trace of it anywhere else on your system (no *.dll files).

If you have administrator’s rights to install software on your computer

Save the file from the Internet or from the CD-ROM if you are in an in-class course by going to the home page and look for the icon **Software** on the navigation panel on the left and click on the icon. In the page that opens, look for **EpiData Entry** and click on the link. In the **File**

download window that opens, click **Save** (not “Open”). Select your directory of choice click **Save**. Repeat the same procedure for **EpiData Analysis**.

In your file manager (Windows Explorer® or an alternative), find now the EpiData Entry setup file:

```
c:\yourfolder\setup_epidata.exe
```

and double-click the file. You will be offered the default installation folder:

```
c:\Program files\EpiData
```

While this is perfectly fine we recommend nevertheless to install it instead in the root:

```
c:\epidata
```

typing over the default that is offered and continue to follow the instructions. Of course, you may install EpiData Entry in the default folder `c:\program files\epidata`, but if you work in a place where installation of software is strictly controlled, this might be precisely the folder that is checked for non-approved program files.

The files will be extracted in a breeze. Among the files you see one named EpiData.exe with the EpiData icon:


EpiData.exe

Right-click on this file, go to **Send To** and **Desktop (create shortcut)** and click the latter. Drag the icon from the desktop down to the task bar. This allows it to be visible from within each program, being just a single click away.

Similarly, install the EpiData Analysis file in the same folder:

```
c:\epidata
```

following the instructions and create a short-cut.

Accessing the EpiData Entry Help file

Windows has changed the way how Help files are being accessed but the necessary plug-in is not part of the operating system in which this change came into effect. As a result, users of Windows Vista or Windows 7 cannot directly access the EpiData Entry Help file and must first visit the Internet, validate that their Windows version is genuine and then download and install the appropriate plug-in for either 32-bit or 64-bit machines. The starting web site is:

<http://support.microsoft.com/kb/917607>

If you do not have administrator’s rights to install software on your computer

Because EpiData software is not interlinked in anyway with the Windows operating system, an actual installation is not really required. You can download the zip file from the EpiData website, and follow the instructions in the `readme.txt` file or ask your facilitator for assistance.

Making a working directory

In order to permit a coordinated activity, all work will be done in one directory / folder.
Create the following folder:

c:\epidata_course

You are now ready to begin the course.

Very handy is to have in addition a temporary folder, such as:

c:\temp

Part A. EpiData Entry

Part A: EpiData Entry

- Exercise 1 A simple questionnaire
- Exercise 2 The QES-REC-CHK triplet
- Exercise 3 Using Labelblocks instead of legal values
- Exercise 4 Adding field labels and value labels
- Exercise 5 Data entry and validation
- Exercise 6 Data safety
- Exercise 7 Using an external file for Labelblocks
- Exercise 8 Dealing with incomplete dates
- Exercise 9 Keeping track of data entry time

Exercise 1: A simple Questionnaire

At the end of this exercise you should be able to:

- a. Define the different types of fields/variables (text, numeric, date) and know when to use them.
- b. Create a data documentation sheet from a simple questionnaire

Like Epi Info 6, EpiData Entry uses the same principle of what we call the QES-REC-CHK (pronounced “Ques-Rec-Check”) files principle.

First we create a questionnaire (a form defining the fields) from which we then create a data entry file, and finally we create a so-called Check file linked to the data entry file to control data entry.

But let us proceed step by step. Let us say we have the following questionnaire:

Laboratory serial number: ____
 Date specimen received (dd/mm/yyyy): ____/____/____
 Sex: ____
 Age in years: ____
 Reason for examination: ____
 Result of specimen 1: ____
 Result of specimen 2: ____
 Result of specimen 3: ____

This might present a typical simple questionnaire as used by an interviewer. Often such questionnaires are first completed on paper. This is actually an excerpt from the Tuberculosis Laboratory Register proposed by The Union:

Tuberculosis Programme

Form 2

Tuberculosis laboratory register

Year _____

Lab Serial No.	Date specimen received	Name	Sex M/F	Age	Name of referring facility	Address - patient for diagnosis	Reason for examination*		Results of specimen			Only for SS+ for diagnosis: TB Number or treatment centre**	Remarks
							Diagnosis (tick)	Month of follow up	1	2	3		

We will use this register as the basis for this course. For the time being, you plan to write a short and concise computer questionnaire, retaining only variables that are easy to capture and are likely to be useful for the analysis.

Each of the questions can be conceived of as a variable and the answer to the question as the value that the variable takes for a particular individual. We will give each variable a unique field name. A completely entered questionnaire for one individual is called a record. We will later enter 15 records (one each for each individual), each with eight fields (corresponding to the number of questions in the questionnaire).

For the time being, we will use field names that consist of one single word that has *not more than ten characters*.

Note that some other analysis software may accept only a field length of eight characters. If you later plan to export your EpiData files for analysis to such a software package and you had used the full field length of ten, then your field names get truncated.

You may verify the set-up in “File” “Options” “Create Data File”. The field name we use might be chosen in a way that it has some meaning relating to the question. There are different types of entry fields for the variables (we will follow the EpiData Entry notation and call them “Fields”):

Text fields: These fields take letters or numbers or a combination of these as possible values, like PETER, KOCH1882, giraffe, 45677 etc. If you enter a number into such a field you will not be able to make any calculation with it. These fields are also sometimes designated as character or alphanumeric fields.

Numeric fields: These are numbers. The numbers might be integers like 885, 33, 1235 or real numbers like 3.4, 6.88, 66.5 (also called floating). You can make calculations with such fields.

Date fields: In different countries, different ways of writing dates are used and this can be confusing for people from another culture. Some write *5 March 2005*, others *March 5 2005*, and again others *2005 March 5*. EpiData Entry lets you choose the type of date you wish to take. In this course we will use European dates, i.e. dates of the format *5 March 2005* or symbolized with DD/MM/YYYY.

One other type of variables is called “logic” or “Boolean” variables. This is sometimes used in food-borne outbreak investigations. There, answers to questions on food items eaten is limited to “yes” and “no” and “missing”. In EpiData Entry, this type of field accepts only the values Y, N, 0, and 1. There is no need for using this additional type of field. It is easy to circumvent by using numeric fields with a label block, and we actually discourage the use of the field type as this is a field type which might pose problems in analysis.

While you are asked to limit the length of the field name, you have much more flexibility with the length of the value a field can take (up to a field length of 80), but we will try to make an as efficient use as possible, that is we will limit the value length to the minimum needed.

Data Documentation Sheet

It is good practice to write what we call a **data documentation sheet** before you make your actual EpiData Entry QES file. EpiData Entry refers to this as **Codebook**.

Note: Field names cannot exceed a length of ten characters, and must be a single word not several words separated by spaces (the space counts). The Field name "Date of birth" would be truncated to "DATE" (which is a reserved name), thus better use "DOB": ensure sticking to single words.

Note: If the Field label begins with a word that is identical to the Field name, you will note later in EpiData Analysis, that this word will be truncated from the Field label. For instance, if your Field name was SEX, and you used SEX OF EXAMINEE as your Field label, this would be truncated to OF EXAMINEE. While this can be fixed easily in EpiData Analysis, it is preferable to prevent it by choosing an alternative Field label during questionnaire design.

This is how we would write such a data documentation sheet:

Field name	Field label	Field type	Field length	Field values	Value labels	Comment
serno	Laboratory serial number *	I	4	1,...,9000 9001, 9002,...		Serial number starting with 1 each year Reserve and assign these numbers sequentially if serial number is not unique, and write a data entry note (use F5 to open a note file)
regdate	Registration date	D	10	01/01/2000,...,31/12/2005 01/01/1800		Range of legal registration dates No or incomplete date provided
sex	Examinee's sex	T	1	F M 9	Female sex Male sex Sex not recorded	

* *Note:* Commonly, it will be preferable to make the identifier a text field. If it is a number, as in this case here with the laboratory serial number, precautions must be taken to distinguish e.g. “0001” from “1”, requiring that the numeric value is entered into one field, and another field, the actual identifier field, is automatically correctly calculated to add leading zeros where appropriate.

Task:

- o *Complete the data documentation sheet for all fields in the questionnaire. Note that you should always define a value if no answer was provided to a question.*

Solution to Exercise 1: A simple Questionnaire

Key Point(s):

- Numbers can be entered into a text field but you will not be able to make any calculations with them.
- It is good practice to write a data documentation sheet before you make your actual EpiData Entry QES file.
- You should always define a value if no answer was provided to a question.
- “Date” is a reserved name in EpiData and cannot be used as a field name.

Task:

- o Complete the data documentation sheet for all fields in the questionnaire. Note that you should always define a value if no answer was provided to a question.*

Solution:

There are many different solutions, but for the sake of uniformity, we will be using the following (but later revise some components of it) as shown on the next page.

Field name	Field label	Field type	Field length	Field values	Value labels	Comment
serno	Laboratory serial number	I	4	1,...,9000 9001, 9002, ...		Serial number starting with 1 each year Reserve and assign these numbers sequentially if serial number is not unique, and write a data entry note (use F5 to open a note file)
regdate	Registration date	D	10	01/01/2000, ..., 31/12/2005 01/01/1800		Range of legal registration dates Date not recorded
sex	Examinee's sex	T	1	F M 9	Female sex Male sex Sex not recorded	
age	Examinee's age in years	I	3	0, ...,125 999		Range of legal years Age not recorded
reason	Examination reason	T	1	D F 9	Diagnosis Follow-up Reason not recorded	
res1	Result of specimen 1	F	3	0.0 1.0 2.0 3.0 4.0 9.0 5.0 6.0 0.1 0.2 0.3 0.4 0.5	Negative 1+ positive 2+ positive 3+ positive 4+ positive No result recorded Positive, not quantified Scanty, not quantified Scanty, 1 AFB per 100 fields Scanty, 2 AFB per 100 fields Scanty, 3 AFB per 100 fields Scanty, 4 AFB per 100 fields Scanty, 5 AFB per 100 fields	

				0.6 0.7 0.8 0.9	Scanty, 6 AFB per 100 fields Scanty, 7 AFB per 100 fields Scanty, 8 AFB per 100 fields Scanty, 9 AFB per 100 fields	
res2	Result of specimen 2	F	3	0.0 1.0 2.0 3.0 4.0 9.0 5.0 6.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9	Negative 1+ positive 2+ positive 3+ positive 4+ positive No result recorded Positive, not quantified Scanty, not quantified Scanty, 1 AFB per 100 fields Scanty, 2 AFB per 100 fields Scanty, 3 AFB per 100 fields Scanty, 4 AFB per 100 fields Scanty, 5 AFB per 100 fields Scanty, 6 AFB per 100 fields Scanty, 7 AFB per 100 fields Scanty, 8 AFB per 100 fields Scanty, 9 AFB per 100 fields	
res3	Result of specimen 3	F	3	0.0 1.0 2.0 3.0 4.0 9.0 5.0 6.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9	Negative 1+ positive 2+ positive 3+ positive 4+ positive No result recorded Positive, not quantified Scanty, not quantified Scanty, 1 AFB per 100 fields Scanty, 2 AFB per 100 fields Scanty, 3 AFB per 100 fields Scanty, 4 AFB per 100 fields Scanty, 5 AFB per 100 fields Scanty, 6 AFB per 100 fields Scanty, 7 AFB per 100 fields Scanty, 8 AFB per 100 fields Scanty, 9 AFB per 100 fields	

Note the following here. For an unknown laboratory date (REGDATE), we must enter a legally existing (valid) date. EpiData will not accept a date 99/99/9999 nor for that matter 29/02/2001. We chose the value “9” for unknown sex, even if we have defined SEX as a character variable and could thus have used “U” (for “unknown sex”). Just note that “9” is treated as a character variable. It is a personal preference of us to usually use 9 or 99.9 or the like to define unknown values, be this for text or numeric variables. We also introduced a “legal range” for some variables like REGDATE and AGE. We did this a bit arbitrarily, but still tried to keep it within what might be expected.

Exercise 2: The QES-REC-CHK triplet

At the end of this exercise you should be able to:

- a. Create and edit a questionnaire file (*.qes).
- b. Make record file (*.rec).
- c. Make and edit a check file (*.chk).

Understand the QES-REC-CHK triplet and how the three are related to each other.

You are now ready to start with the design of the questionnaire in EpiData Entry, based on your data documentation sheet.

Open EpiData Entry by double-clicking the icon on your desktop or single-clicking the icon in your quick-launch task bar. You see the EpiData Entry task bar on top of the screen. It has three rows. For the time being we concentrate only on the middle row that shows the following sequence (this is called the “Process bar”):



Each of these has a menu which you see when you click on the box. You can see immediately where you have to start.

Step 1: Creating the *.QES file

If you click on “**1. Define data**” (or using the shortcut **ALT+1**) the menu with two options pops up: “New .QES file” and “Open .QES file”. EpiData questionnaire files have the extension “*.QES”. As you must now create a questionnaire file, click “New .QES file” and the empty screen ready for writing opens.

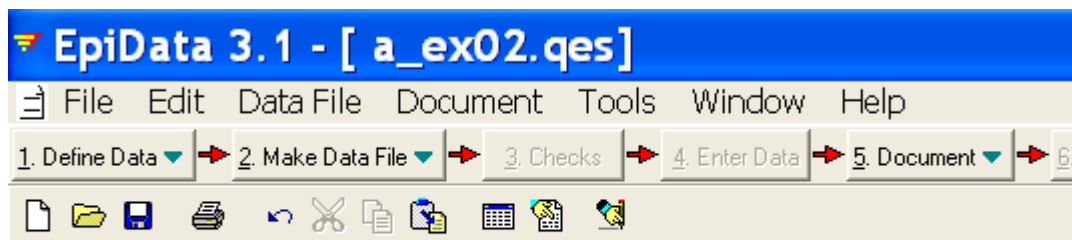
Start to type like in any word processor the following:

```
This is the questionnaire for the laboratory register
```

```
serno
```

Let’s save this right away as **A_EX02.QES** (shortcut: **CTRL+S**).

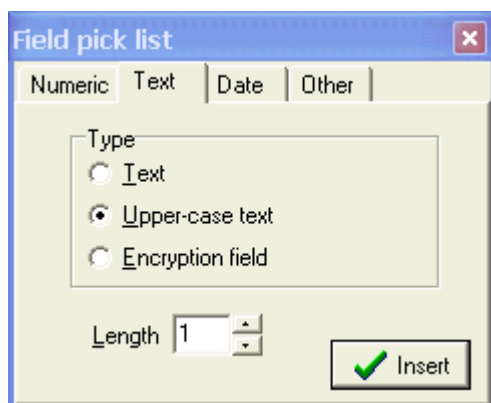
We have to associate now two things with this field, the type of field and the field length. In the top menu line you see “Edit”.



This is the questionnaire for the laboratory register
 serno |

Note: While EpiData Entry is not case-sensitive (that is, you may use upper-case or lower-case), some statistical packages are case-sensitive (“sex” is not identical to “Sex” or “SEX”). You are on the safe side if you make it a habit of using always lower-case for field names. See also “File” “Options” “Create Data File” to force lower-case in the data file.

Click on it to get the drop down menu. However, we encourage you to learn the shortcuts without using the mouse. If you click the **Alt** key, you see that each of these menus has one letter underlined. In this case it is the E in Edit. Thus **Alt+E** is a fast way to see the drop down menu. In it, you see “Field pick list **CTRL+Q**”. Pick it by typing “f” twice in sequence and the Field pick list box opens. Our field is a text field. You see that there are three types:



Text: using this type you may enter upper or lower case letters or a mixture of cases, and the value will be exactly as you entered it.

Upper-case text: if you enter a lower-case letter from your keyboard, it will automatically be converted to upper case. This is often very useful as the field value “m” is not the same as “M” and we would get a counting for each of these values depending of the data entry person’s preference. *We will thus use this option for now.*

Encryption field: this is a powerful tool if you enter information like personal identifiers that should be openable only by persons with access to a password. More on data encryption later in this course.

Choose now Numeric and a field length of 4. Make sure that you have a space between the field name and the field type / field length:

```
serno ####
```

You will complete the questionnaire later in the Tasks. For the time being, let's leave it with just this one field and save the questionnaire as A_EX02.QES (the extension .QES is automatically supplied).

2. Make Data File ▼

Step 2: Making the *.REC file

We have now the QES file, and this provides the information on field definitions for the data file. **Alt+2** opens the drop down menu to pick the Make data file sub-menu which opens the “Create data file from *.QES file” dialogue box. The name of the *.QES file is already there and the same file name (with the *.REC extension) is proposed. That is very sensible, it should be so, and it must be so, and we accept this. We are now prompted to enter a description. This is not necessary but it is helpful documentation. Thus we type in for instance “Exercise 2” and we are informed that the A_EX02.REC has been created.

3. Checks

Step 3: Making the *.CHK file

With **Alt+3** we open the dialog box “Select data file for checks”. The A_EX02.REC is suggested and that is the file we need. Thus open it. The entry screen appears with one variable and a box showing that we have indeed now the third file A_EX02.CHK.

It shows the field SERNO and that this is a Numeric field. It also shows different things we could modify for the checks of this field:

Range Here you can enter all legal values, i.e., the Range defines what values the data entry person is allowed to enter. In the case of the unique identifier, there will, by definition, be as many as there are individuals on whom information has to be entered. It does therefore not make any sense to define legal values for the variable SERNO. Examples of legal values for the variables SEX and REGDATE could be:

```
M, F, 9  
01/01/2000-31/12/2005, 01/01/1800
```

Jumps You could determine here if a subsequent field should be skipped if the current value takes a certain value. For instance, you may have the value “M” for male and the value “F” for female in the field SEX. If the person is female, you might ask how many pregnancies (variable, e.g., PREGNO) she has had, but if the person is male, you would obviously not ask this question. Thus, you would jump (bypass) the question about pregnancies in case the study subject is male and go in that case to the field after pregnancies which might be AGE. To tell that a jump is needed in case the value of the field SEX is “M”, you would simply type:

M>age

In the case of the field SERNO, no Jumps are needed, and we will leave this open.

Must enter Here you define whether information must be entered or not. If you state that it must be entered, values for the following field cannot be entered unless the current one has a value entered. In the case of the field SERNO, we will require that it is entered. Choose thus “yes” from the drop-down menu at the right. You will therefore not have any missing values for this variable. **Note: in this course, we will always use Must enter fields** (except for automatically calculated variables).

Repeat Here you can specify whether a value for the field should be repeated in all subsequent records, unless you choose to overwrite. This comes in handy when you make a file for a specific district, and the district name in this file is always the same. For the field SERNO this is obviously not the case.

Value label If we code, e.g., the value “female” for the field SEX with “1” and the value “male” with “2”, then it could prove useful to label it so that an explanation appears that “1” stands for “female” and “2” for “male” to reduce data entry errors. This will be shown in the next exercise. It will make the life of the data entry person so much easier.

You are reminded to save by clicking on ‘save and close’ after working on checks.

Tasks:

- o Open the existing A_EX02.QES file and complete it.*
- o Create the A_EX02.REC file (overwrite the existing one).*
- o Edit the A_EX02.CHK file, make all fields Must enter fields, and try to enter legal values where appropriate (and as defined in the data documentation sheet).*

Solution to Exercise 2: The QES-REC-CHK triplet

Key Point(s):

- For text fields it is frequently better to make them 'upper-case text' as 'm' is not the same as 'M'.
- In the CHK file, it is always better to make all fields 'Must enter', except for automatically calculated variables. Then you will not have any missing values for the variable.

Tasks:

- o *Open the existing A_EX02.QES file and complete it*
- o *Create the A_EX02.REC file (overwrite the existing one)*
- o *Edit the A_EX02.CHK file, make all fields Must enter fields, and try to enter legal values where appropriate (and as defined in the data documentation sheet).*

Solution:

The questionnaire would look as follows:

This is the questionnaire for the laboratory register

```
serno ####
labdate <dd/mm/yyyy>
sex <A>
age ###
reason <A>
res1 #.#
res2 #.#
res3 #.#
```

EpiData Entry software is not case-sensitive, but the definition of values for character fields obviously is. Nevertheless, you make it best a habit to use lower-case for field names as some statistical packages are case-sensitive when it comes to field names.

The questionnaire looks a bit ragged and it will be so in the data entry form as well, but there is an easy way to line it up properly. Choose the field with the longest name. Here, four have the same length, thus go anywhere on the line with the field REGDATE, choose "Edit" "Align fields" and you get the following much nicer outline:

This is the questionnaire for the laboratory register

```
serno      #####
regdate    <dd/mm/yyyy>
sex        <A>
age        ###
reason     <A>
res1       #.#
res2       #.#
res3       #.#
```

If the QES file is open, use **CTRL+T** to look how the data entry from will look:

This is the questionnaire for the laboratory register

serno	<input type="text"/>
regdate	<input type="text"/>
sex	<input type="text"/>
age	<input type="text"/>
reason	<input type="text"/>
res1	<input type="text"/>
res2	<input type="text"/>
res3	<input type="text"/>

The *.REC file we created can be looked at in a text editor (like NotePad™ that comes with Windows™) and we see the following:

```
9 1 VLAB Filelabel: Exercise 2: The QES-REC-CHK triplet
_label11      1  1 30  0  0  0  0 112 This is the questionnaire for the laboratory register
#serno        1  3 30 11  3  0  4 112 serno
_regdate      1  4 30 11  4 11 10 112 regdate
_sex          1  5 30 11  5  3  1 112 sex
#age          1  6 30 11  6  0  3 112 age
_reason       1  7 30 11  7  3  1 112 reason
#res1         1  8 30 11  8 101  3 112 res1
#res2         1  9 30 11  9 101  3 112 res2
#res3         1 10 30 11 10 101  3 112 res3
```

While this is perhaps not very informative to you at this point in time, you may note the simplicity of it. Have you ever tried to look at a spreadsheet in a text editor? You cannot, as it will not load and all you see is some gibberish. In contrast this is a straight simple text file and its file size is just 613 bytes. In comparison, a Microsoft Word® 1997-2003 file containing the single letter “a”, nothing else, weighs in at 24,576 bytes...

For the *.CHK file, the entering of ranges and legal values took perhaps a bit trial and error. But basically it is very simple. For the field REGDATE we just entered:

```
01/01/2000-31/12/2005,01/01/1800
```

and for AGE

0-120,999

We can open the A_EX02.CHK file (CTRL+O, “Files of type”, “EpiData check file (*.chk)”) it is just a text file after all. It looks as follows:

```
serno
  RANGE 1 9999
  MUSTENTER
END
```

```
regdate
  RANGE 01/01/2000 31/12/2005
  LEGAL
    01/01/1800
  END
  MUSTENTER
END
```

```
sex
  LEGAL
    M
    F
    9
  END
  MUSTENTER
END
```

```
age
  RANGE 0 125
  LEGAL
    999
  END
  MUSTENTER
END
```

```
reason
  LEGAL
    D
    F
    9
  END
  MUSTENTER
END
```

```
res1
  RANGE 0.0 1.0
  LEGAL
    2.0
    3.0
    4.0
    5.0
    6.0
    9.0
  END
  MUSTENTER
END
```

```
res2
  RANGE 0.0 1.0
  LEGAL
```

```
    2.0
    3.0
    4.0
    5.0
    6.0
    9.0
  END
  MUSTENTER
END

res3
  RANGE 0.0 1.0
  LEGAL
    2.0
    3.0
    4.0
    5.0
    6.0
    9.0
  END
  MUSTENTER
END
```

Note that the capitalization versus the use of lower-case letters is used here only for easier visualization of the program flow (see note above about EpiData not being case-sensitive).

We will learn very soon how to edit the *.CHK file directly to experience its tremendous power.

Exercise 3: Using Labelblocks instead of legal values

At the end of this exercise you should be able to:

- Change fields by editing *.qes and *.chk files
- Use Labelblocks instead of legal values

The use of legal values is a powerful tool to control data entry errors. It is also perfectly fine for the beginner to take recourse to these and if you enter the data by yourself. Nevertheless, if another person is entering the data, you can assist so that the life of that person becomes much easier. However, this requires you to prepare the Check file to make it much more convenient as you will see.

In Exercise 2 you have made the A_EX02.CHK file, where we currently have:

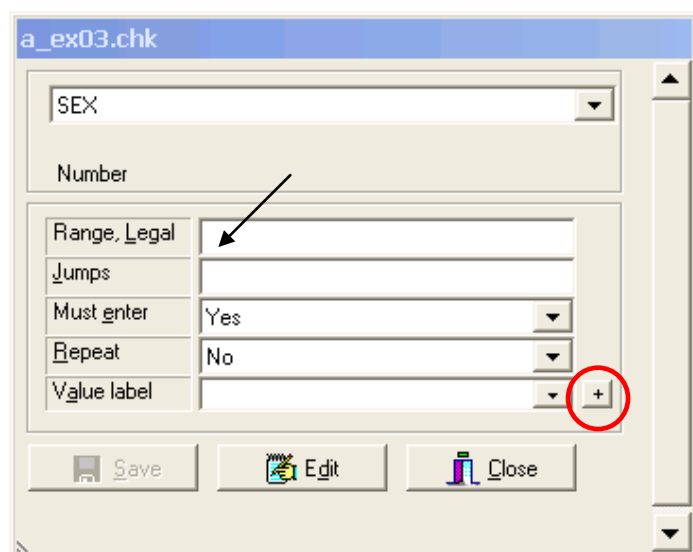
```
sex
  LEGAL
    F
    M
    9
  END
MUSTENTER
END
```

This will need to be changed if instead of legal values we are going to use “Labelblocks” (henceforth used as a single word like in the language that EpiData Entry requires): **one should never use both, legal values and invoking a Labelblock.**

We can accomplish both at the same time interactively with **ALT+3** (Add / revise Checks):

This is the questionnaire for the laboratory register

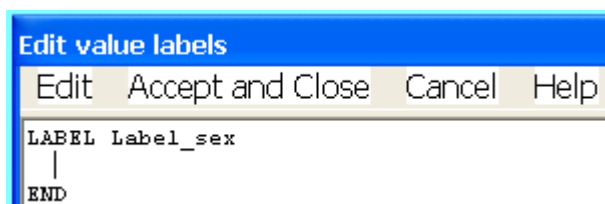
serno	<input type="text"/>
regdate	<input type="text"/>
sex	<input type="text"/>
age	<input type="text"/>
reason	<input type="text"/>
res1	<input type="text"/>
res2	<input type="text"/>
res3	<input type="text"/>



removing the legal values first. At the bottom line Value label you see at the left the “+” sign:



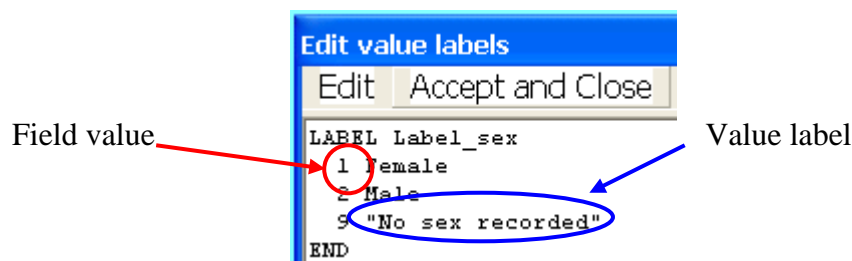
If you click this, the Check file opens and offers the block for the label and a name for it:



with the cursor ready for entering your content. As we prefer numeric coding we enter the Field value as a number and the Value label as an explanatory text:

1 Female

Note that the Value label Female is not within quotation marks, while the Value label "No sex recorded" is:



Quotation marks are required if the Value label consists of two or more words separated by a space. However, always using quotation marks is also legal.

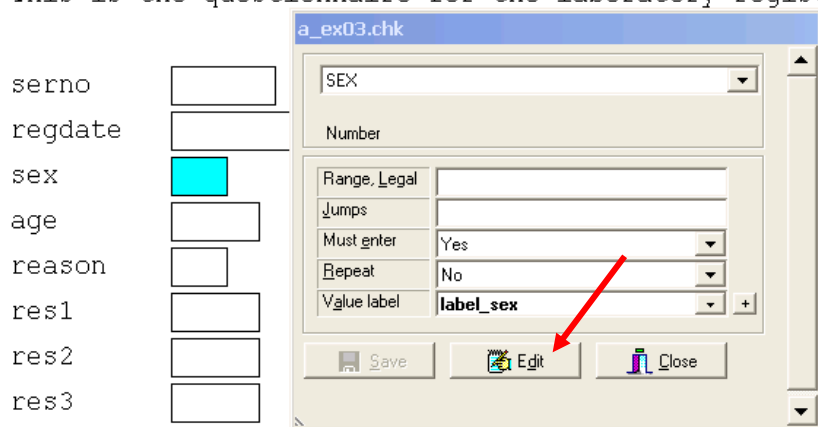
Accept and close save the Check file and close the Check file editor box.

Editing the CHK file

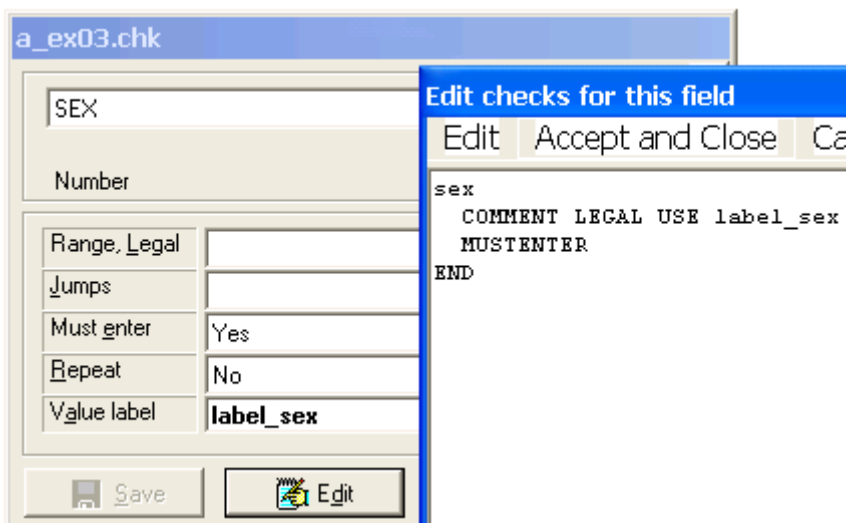
There are two ways to make additional changes in your CHK file:

1) Use **Edit**:

This is the questionnaire for the laboratory register



If you open **Edit** for the field SEX:



you get the information on the field SEX that is written into the CHK file displayed and you can edit it here directly. This has the distinct advantage that after you have made a change and “Accept and Close” you get right away a response whether what you did is accepted as legal by EpiData Entry. A disadvantage is that you see only the information on the field on which the focus currently is. Advanced editors choose thus sometimes the second approach:

2) Opening the CHK file

You open the CHK file from the File menu and you see the entire CHK file. In the following only the part before and after the field SEX is shown:

```
regdate
  RANGE 01/01/2000 31/12/2005
  LEGAL
    01/01/1800
  END
  MUSTENTER
END

sex
  COMMENT LEGAL USE label_sex
  MUSTENTER
END

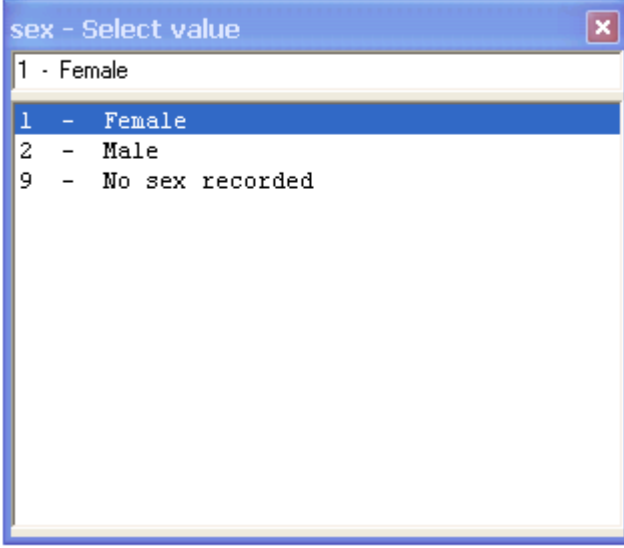
age
  RANGE 0 125
  LEGAL
    999
  END
  MUSTENTER
END
```

While there is often an advantage to see the whole CHK file (and you will have to do that if there are errors), the disadvantage is that you can save an erroneous CHK file in this mode and any logical error in structuring it or misspelling of commands will be reported only once you try to enter data.

Whichever approach you choose, you will need to edit the commands for the field SEX. If you leave it as it is, the Labelblock will not be displayed during data entry. To make it being displayed as follows:

This is the questionnaire for the laboratory register

serno	<input type="text" value="1432"/>
regdate	<input type="text" value="20/03/2003"/>
sex	<input type="text" value="1"/>
age	<input type="text"/>
reason	<input type="text"/>
res1	<input type="text"/>
res2	<input type="text"/>
res3	<input type="text"/>



you have to append it with the command SHOW:

```
sex
COMMENT LEGAL use label_sex SHOW
MUSTENTER
END
```

Tasks:

- o **Open Add / revise Checks (ALT+3), and make the necessary changes for the fields SEX, REASON, RES1, RES2, RES3.**
- o **Save the A_EX02.QES file as A_EX03.QES and edit it to make the fields SEX and REASON numeric, then make a new A_EX03.REC file based on this QES file.**

Solution to Exercise 3: Using Labelblocks instead of legal values

Key Point(s):

- One should never use both, legal values and invoking a Labelblock
- When Labelblocks are used with numeric coding, the label linked to a numeric value will always be shown.

Tasks:

- o *Open Add / revise Checks (ALT+3) and take the necessary changes for the fields SEX, REASON, RES1, RES2, RES3.*
- o *Save the A_EX02.QES file as A_EX03.QES and edit it to make the fields SEX and REASON numeric, then make a new A_EX03.REC file based on this QES file.*

Solution:

This is the Check file A_EX03.CHK:

```
LABELBLOCK
  LABEL label_sex
    1 Female
    2 Male
    9 "No sex recorded"
  END
  LABEL label_reason
    0 Diagnosis
    8 "Follow-up, month not stated"
    9 "Reason not stated"
    1 "Follow-up at 1 month"
    2 "Follow-up at 2 months"
    3 "Follow-up at 3 months"
    4 "Follow-up at 4 months"
    5 "Follow-up at 5 months"
    6 "Follow-up at 6 months"
    7 "Follow-up at 7 months or later"
  END
  LABEL label_result
    0.0 Negative
    1.0 "1+ positive"
    2.0 "2+ positive"
    3.0 "3+ positive"
    4.0 "4+ positive"
    9.0 "No result recorded"
    5.0 "Positive, not quantified"
    6.0 "Scanty, not quantified"
    0.1 "Scanty, 1 AFB per 100 fields"
    0.2 "Scanty, 2 AFB per 100 fields"
    0.3 "Scanty, 3 AFB per 100 fields"
    0.4 "Scanty, 4 AFB per 100 fields"
    0.5 "Scanty, 5 AFB per 100 fields"
    0.6 "Scanty, 6 AFB per 100 fields"
    0.7 "Scanty, 7 AFB per 100 fields"
    0.8 "Scanty, 8 AFB per 100 fields"
```

```

    0.9 "Scanty, 9 AFB per 100 fields"
END
END

serno
  LEGAL 1 9999
  MUSTENTER
END

regdate
  RANGE 01/01/2000 31/12/2005
  LEGAL
    01/01/1800
  END
  MUSTENTER
END

sex
  COMMENT LEGAL USE label_sex SHOW
  MUSTENTER
END

age
  RANGE 0 125
  LEGAL
    999
  END
  MUSTENTER
END

reason
  COMMENT LEGAL USE label_reason SHOW
  MUSTENTER
END

res1
  COMMENT LEGAL USE label_result SHOW
  MUSTENTER
END

res2
  COMMENT LEGAL USE label_result SHOW
  MUSTENTER
END

res3
  COMMENT LEGAL USE label_result SHOW
  MUSTENTER
END

```

Quite obviously, the ease of data entry using Labelblocks makes them an attractive alternative to legal values whenever this is possible (but it is not always feasible).

One advantage of Labelblocks is that we have no confusion if we use numeric coding because the label linked to a numeric value will always be shown. From now on we will be using numeric coding whenever possible.

Exercise 4: Adding field labels and value labels

At the end of this exercise you should be able to:

- a. Add a Field label to a Field name in the questionnaire
- b. Edit the CHK file to show the Value and Value labels
- c. Ensuring that the identifier is unique

Field labels and value labels

Our questionnaire shows now the short Field name and the box for the Field value. This is perhaps sufficient if we enter the data ourselves but it might be difficult for a data entry person not familiar with the definition of the fields. EpiData Entry allows the addition of Field labels to the Field name which become part of the Field definition. As the Field values are numeric, i.e., for instance “1” to indicate “Female sex”, the data entry person does not see after entry whether the value that was entered was really the value intended. It is possible to show the Value label to the left of the Value. Adding these two components to the QES (the Field label) and the CHK file (the instruction to show the Value label) is part of this exercise. In summary, what this exercise is to accomplish is to have the following display shown here for the Field SEX:

Field name	Field label	Value	Value label
sex	Sex of examinee	9	No sex recorded

The Field label is added to the questionnaire, while the instruction to display the Value label is added to the Check file with TYPE COMMENT as follows:

```
sex
  COMMENT LEGAL USE label_sex SHOW
  MUSTENTER
  TYPE COMMENT
END
```

Ensuring that the identifier is unique

The field SERNO will later be used as a *unique identifier*. That it is, is relatively easily to ascertain in the particular case of the laboratory register where the serial number is entered sequentially. But it would be impossible to know whether a particular identifier had not been used before if identifiers are codes of the type AX7, ZV4, YY3, etc, and in no particular sequence. It is possible to tell EpiData Entry to check whenever an identifier is entered whether it has ever been used before and is thus not unique. The command that needs to be added is:

```
serno
  MUSTENTER
  KEY UNIQUE
END
```

or

```
serno
  MUSTENTER
  KEY UNIQUE 1
END
```

The command KEY in a given field results in the writing of another file with the extension *.EIX, a so-called index file. Index files allow a very rapid search for specific information. Thus if the field SERNO is a KEY UNIQUE field, every time a serial number is entered, the entire database is searched whether this registration number is truly unique. If it is not, a warning will inform the data entry person that the identifier had been used before and in which record and whether the person wants to see that record. It will be impossible to continue to enter data without entering a truly unique identifier.

The number 1 in KEY UNIQUE 1 above is optional. If not used, internally KEYS are numbered sequentially up to the maximum allowable 10 KEY fields that can be used in a record.

If a data entry person uses the mouse to bypass the field SERNO then the command KEY UNIQUE cannot be executed and it is possible to save the record to disk without the identifier. This will be possible without any problem for 1 record, but will pose a problem during validation which will be on the KEY. If an attempt is made to save a second record without SERNO, the data entry person will be warned that the KEY is not unique. This is so because the missing value is represented by a period (.) and having twice a period will make it non-unique. While it is bad to have missing values in MUSTENTER fields (indicating that the mouse was used), it very troublesome not to have a unique identifier.

The best strategy is thus to prevent the possibility to save a record without an identifier. We add thus to the CHK file an AFTER RECORD command that checks whether the identifier is present, to issue a warning if not, and get the data entry person back to the identifier field:

```
AFTER RECORD
  IF serno=. THEN
    HELP "Core information missing:\n SERNO\n must be available" TYPE=WARNING
    GOTO serno
  ENDIF
END
```

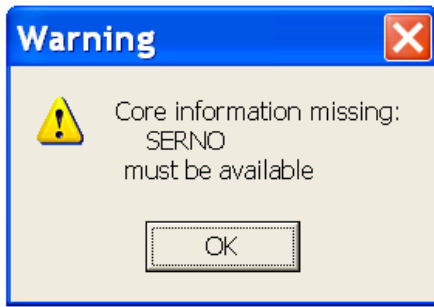
The command AFTER RECORD concluded by the command END contains nested commands that are executed before the record can be saved. It checks whether the field SERNO has missing information (“.”), and if so will take the data entry person back to the field SERNO with the following:

```
IF serno=. THEN
  GOTO serno
ENDIF
```

In order to inform the data entry person why the cursor will go back to the field SERNO, the command HELP is added. As it is written above, the line:

```
HELP "Core information missing:\n SERNO\n must be available" TYPE=WARNING
```

Will result in the display of a WARNING box:



Task

- o Revise the A_EX03.QES file, save it as A_EX04.QES, make the REC file, open the A_EX03.CHK file, save it as A_EX04.CHK and make the necessary changes and then enter some made-up data to check the functionality.***

Solution to Exercise 4: Adding field labels and value labels

Key Point(s):

- A Field label helps the data entry person to become familiar with the definition of the fields.
- A Value label helps the data entry person to see Field values and Value labels. This is particularly helpful when the Field values are numeric.

Task

- o *Revise the A_EX03.QES file, save it as A_EX04.QES, make the REC file, open the A_EX03.CHK file, save it as A_EX04.CHK and make the necessary changes and then enter some made-up data to check the functionality.*

Solution

The data entry form looks as follows (**CTRL+T**):

This is the questionnaire for the laboratory register

serno	Laboratory serial number	<input type="text"/>	Write note (F5) if alternative number must be used
regdate	Registration date	<input type="text"/>	Enter 01/01/1800 if not recorded
sex	Examinee's sex	<input type="text"/>	
age	Examinee's age in years	<input type="text"/>	Enter 999 if not recorded
reason	Examination reason	<input type="text"/>	
res1	Result of specimen 1	<input type="text"/>	
res2	Result of specimen 2	<input type="text"/>	
res3	Result of specimen 3	<input type="text"/>	

You note that in this solution we have added more to the QES file than was asked for in the **Task**: information was written into the QES file to the left of the field, but only for some fields.

When to write instructions into the QES file?

The fields SEX, REASON, RES1, RES2, and RES3 all have Labelblocks. Thus, all the instructions what can be entered are provided. Furthermore, the Value label is written into the space right to the field. There is thus no need to write an instruction nor would it be sensible to do so. On the other hand, the fields SERNO, REGDATE, and AGE do not have Labelblocks and the data entry person will thus not know (before making an error when the message says what is legal) what to do if there is a duplicate SERNO (a non-unique identifier) or if the registration date is missing. It is thus a good idea to provide for fields which do not have Labelblocks instructions directly in the QES file.

And this is the A_EX04.CHK file (**CTRL+O**, pick EpiData check files as Type of files):

```
LABELBLOCK
  LABEL label_sex
    1 Female
```

```

    2 Male
    9 "No sex recorded"
END
LABEL label_reason
    0 Diagnosis
    8 "Follow-up, month not stated"
    9 "Reason not stated"
    1 "Follow-up at 1 month"
    2 "Follow-up at 2 months"
    3 "Follow-up at 3 months"
    4 "Follow-up at 4 months"
    5 "Follow-up at 5 months"
    6 "Follow-up at 6 months"
    7 "Follow-up at 7 months or later"
END
LABEL label_result
    0.0 Negative
    1.0 "1+ positive"
    2.0 "2+ positive"
    3.0 "3+ positive"
    4.0 "4+ positive"
    9.0 "No result recorded"
    5.0 "Positive, not quantified"
    6.0 "Scanty, not quantified"
    0.1 "Scanty, 1 AFB per 100 fields"
    0.2 "Scanty, 2 AFB per 100 fields"
    0.3 "Scanty, 3 AFB per 100 fields"
    0.4 "Scanty, 4 AFB per 100 fields"
    0.5 "Scanty, 5 AFB per 100 fields"
    0.6 "Scanty, 6 AFB per 100 fields"
    0.7 "Scanty, 7 AFB per 100 fields"
    0.8 "Scanty, 8 AFB per 100 fields"
    0.9 "Scanty, 9 AFB per 100 fields"
END
END

serno
    MUSTENTER
    KEY UNIQUE
END

regdate
    RANGE 01/01/2000 31/12/2005
    LEGAL
        01/01/1800
    END
    MUSTENTER
END

sex
    COMMENT LEGAL USE label_sex SHOW
    MUSTENTER
    TYPE COMMENT
END

age
    RANGE 0 125
    LEGAL
        999
    END
    MUSTENTER
END

reason
    COMMENT LEGAL USE label_reason SHOW
    MUSTENTER
    TYPE COMMENT
END

```

```
res1
  COMMENT LEGAL USE label_result SHOW
  MUSTENTER
  TYPE COMMENT
END

res2
  COMMENT LEGAL USE label_result SHOW
  MUSTENTER
  TYPE COMMENT
END

res3
  COMMENT LEGAL USE label_result SHOW
  MUSTENTER
  TYPE COMMENT
END

AFTER RECORD
  IF serno=. THEN
    HELP "Core information missing:\n      SERNO\n must be available" TYPE=WARNING
    GOTO serno
  ENDIF
END
```

Exercise 5: Data entry and validation

At the end of this exercise you should be able to:

- Know the three ways of reducing data entry errors
- Copy the structure of a REC file
- Export data from EpiData files
- Validate duplicate data files

You have a line listing of 15 records on the pages following the task. The data from these should now be entered. But before you start working, a few considerations are in place.

Ensuring quality data entry

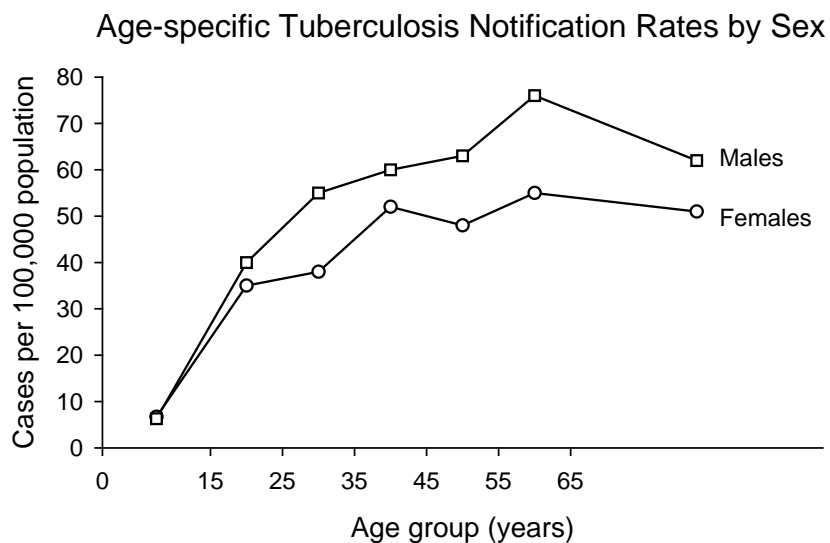
The motto for this course is:

“You wish never to find yourself in a position to defend the quality of your data”

Michael B Gregg, formerly MMWR Editor, deceased

You might be challenged about the interpretation of your data, that is part of the scientific process, but your data should be of impeccable quality.

What do you think about the following graph?



It looks nice and we could talk about the differences between males and females and this and that. But we will keep it short: it is total nonsense. The data underlying this graph have no basis, they were made up. Of course, if we were to present these data for real, it would be outright scientific fraud. Few people commit that (but it exists). **Nevertheless, often no assurance can be given that the computerized data are a true reflection of the original data source.** People may have in all honesty done “their best” and assume that they made no errors or so few that it really doesn’t matter. However, this is not good enough for science in general and public health and epidemiology in particular.

There are three ways how we reduce and ultimately eliminate data entry errors:

- o Using a *.CHK file
- o Working together
- o Duplicate data entry and validation

Using a *.CHK file

We have already a few conditions inbuilt that limit data entry errors by creating the A_EX04.CHK file. For instance, a MUSTENTER field will prevent a data entry person to skip an actually recorded value, as one cannot continue without having entered a value for that field. For the field SEX, we allowed only 1, 2, and 9 as legal values. It is thus not possible to enter “3” into this field. Combined with the pop-up menu during entry, no confusion can arise. The *.CHK file is an extremely powerful tool to control how data are entered.

Working together

Entering data alone requires shifting attention constantly between the paper record and the computer screen. This will almost by necessity result in numerous errors, being it that one record is skipped or that it is forgotten what we just read. It should be routine that two persons work on data entry: one person reads aloud the Field value, the other repeats it aloud and enters the value.

Duplicate data entry and validation

Even with both of the above precautionary measures, data entry errors will still occur, and worse, to an unknown extent. ***The only way, and the only acceptable one, is to enter the data twice into two different files, and then to compare the two files for discordances.*** Any discordance uncovered will then be corrected against the original paper record.

EpiData Entry provides this powerful tool and it offers two approaches to it. The first approach is to enter the data independently twice. The second approach is to prepare for duplicate entry. After the first file is completed, the second file is prepared based on a key field for the first file. While then entering the second duplicate file, the value is checked for each field in each record against the same record of the first file while entering it and you are warned of any discordance, so that you can ensure proper recording during the second entry process.

In either case, we need a unique identifier. We have made a provision that we have such an identifier (see Exercise 4). Sometimes, it must be constructed from more than one variable, an approach you are going to learn later. Laboratory numbers are serial, and it is thus usually seen whether they are unique, but with some other identifiers this is not the case. We will

thus use an EpiData Entry command that will ensure that EpiData checks every “unique identifier” whether it is really unique. To the field for the serial number we thus added the following command in the previous exercise:

```
serno
  MUSTENTER
  KEY UNIQUE 1
END
```

If a duplicate key is revealed, then a data entry note (a *.not file) should be written. This can be invoked with **F5**. In this note, you would exactly specify with what identifier you have replaced the duplicate key, so that this note can be given to those who enter the data the second time, enabling them to use the same alternative key.

At this point in time, you will be using the first approach, and that is to independently enter the 15 records twice and then to compare the two files.

Note for data entry: Do never move around the fields with the help of the mouse. The mouse movements can not be recorded properly and unforeseen errors may occur (e.g., bypassing a calculation made in a field, missing a MUSTENTER command, etc), because the Check file cannot be applied to fields you skip by moving the mouse from one to another. Use only TAB, cursor keys and the Enter key to move around an EpiData entry form.

Ensuring that we have a unique identifier before saving the record to disk

We can do all pleading (above) not to use the mouse or not to save an unfinished record, but data entry persons are bound to perhaps defeat all pleas. It is thus best to do something in the CHK file that makes it failsafe. While one would hope that the mouse is never used to skip a field, if it is done, then one has missing information for the field which is bad. But not having a unique identifier is close to disastrous as one would see when validating data. The following commands at the end of the CHK file will prevent the data entry person to save a record without a unique identifier (SERNO in this case) (see end of previous exercise):

```
AFTER RECORD
  IF serno=. THEN
    HELP "Core information missing:\n SERNO\n must be available" TYPE=WARNING
    GOTO serno
  ENDIF
END
```

Tasks:

- o **Take your A_EX04.REC file, go to “Tools” “Copy structure” and copy the A_EX04.REC including its A_EX04.CHK file to:
A_EX05 A.REC and A_EX05A.CHK files
A_EX05 B.REC and A_EX05B.CHK files**
- o **Enter the 15 records using the A_EX05A.REC file.**

After completing data entry, enter the same data again into to the A_EX05B.REC file.

- o After you have completed the two files, go to “5. Document” “Validate Duplicate Files” and produce a *.NOT file giving you a list of any discordance. Save the *.NOT file as A_EX05AB.NOT*
- o Use “6. Export Data” “Epidata” to export either one of the two files to a new A_EX05F.REC file, and then make all corrections in this file. This is your final dataset.*

On the next page you find the dataset with 15 records

Laboratory: Ganda Chivua

Tuberculosis laboratory register

Year: 2002

Lab Serial No.	Date specimen received	Name	Sex M/F	Age	Name of referring facility	Address - patient for diagnosis	Reason for examination*		Results of specimen			Only for SS+ for diagnosis: TB Number or BMU**	Remarks
							Diagnosis (tick)	Month of follow up	1	2	3		
3298	26 Oct	Mary	F	35	Bindura	Beijingstr. 6		5	neg	neg			
3299	26 Oct	John	M	20	Awuna	Tokyo Ave 5	√		neg	neg	neg		
3300	26 Oct	Petra	F	30	Birchenough	Bangkok Rd 108		5	neg	neg			
3301	26 Oct	Charles	M	24	Bindura	Hanoi Street 7a		2	neg	neg			
3302	26 Oct	Tiffany	F	38	Bindura	Hongkong Ave 8	√		neg	neg	neg		
3303	26 Oct	George	M	60	Bindura	Zurich Rd 923	√		neg	neg	neg		
3304	26 Oct	Luke	M	78	Awuna	Paris Street 18a	√		neg	neg	neg		
3304	26 Oct	Virginia	F	28	Birchenough	London Rd 24	√		neg	neg	neg		
3305	27 Oct	David	M	50	Awuna	Baltimore Str 1		6	neg	neg			
3306	27 Oct	Hans	M	50	Ganda Chivua	Bern Str 12	√		1+	1+	1+	Ganda Chivua No 342	
3307	27 Oct	Bill	M	68	Bindura	Berlin Ave 88	√		neg	neg	neg		
3308	27 Oct	Susan	F	29	Birchenough	Amsterdam Rd 3		5	neg	neg			
3309	27 Oct	Marc	M	36	Bindura	Vienna Str 76		2	neg	neg			
3310	27 Oct	Eve	F	15	Awuna	Rome Ave 4		5	neg	neg			
3311	27 Oct	Anthony	M	37	Birchenough	Antwerp Str 26c		6	neg	neg			

* Check the appropriate category from the *Request for Sputum Examination*

**TB register number or name of the referral BMU (Basic Management Unit)

Solution to Exercise 5: Data entry and validation

Key Point(s):

- It should be routine that two persons work on data entry, and never one.
- The only and acceptable way to minimize data entry errors, is to enter the data twice into two different files, and then compare the two files for discordances.
- Never use the mouse to move around fields during data entry, because the Check file cannot be applied to fields you skip by moving the mouse from one field to another.

Tasks:

- o *Take your A_EX04.REC file, go to “Tools” “Copy structure” and copy the A_EX04.REC including its A_EX04.CHK file to:
A_EX05 A.REC and A_EX05A.CHK files
A_EX05 B.REC and A_EX05B.CHK files*
- o *Enter the 15 records using the A_EX05A.REC file.*

After completing data entry, enter the same data again into to the A_EX05B.REC file.

- o *After you have completed the two files, go to “5. Document” “Validate Duplicate Files” and produce a *.NOT file giving you a list of any discordance. Save the *.NOT file as A_EX05AB.NOT*
- o *Use “6. Export Data” “Epidata” to export either one of the two files to a new A_EX05F.REC file, and then make all corrections in this file. This is your final dataset.*

Solution:

Depending on the errors you made, you will get an output like the following:

```
VALIDATE DUPLICATE DATA FILES REPORT
=====
```

```
Report generated 21. Jun 2008 15:45
```

```
-----
Data file 1:
```

```
-----
File name:      C:\epidata_course_Mongolia\course_exercise_files\a_ex05a.rec
File label:     Exercise 5: Data entry and validation
File date:      21. Jun 2008 15:44
Records total: 15
```

```
-----
Data file 2
```

```
-----
File name:      C:\epidata_course_Mongolia\course_exercise_files\a_ex05b.rec
File label:     Exercise 5: Data entry and validation
File date:      21. Jun 2008 16:01
Records total:15
-----
```

```
Options for validation:
Ignore deleted records:          Yes
Ignore text fields:              No
Ignore letter-case in text fields: No
Report differences in field types: No
Ignore missing records in data file 2  No
```

```
Fields in both data files that were used in the validation:
SERNO,REGDATE,SEX,AGE,REASON,RES1,RES2,RES3
```

```
Fields excluded from data file 1:
None
```

```
Fields excluded from data file 2:
None
```

```
Fields used as index keys:
SERNO
```

```
-----
RESULTS OF VALIDATION:
-----
```

```
Records missing in data file 1:      0
Records missing in data file 2:      0

Number of common records found:      15
Number of fields checked per record:  8
Total number of fields checked:      120
```

```
2 out of 15 records had errors (13.33 pct.)
2 out of 120 fields had errors ( 1.67 pct.)
```

```
-----
DATA FILE 1 | DATA FILE 2
-----
Record key field(s): (Rec. # 11) | Record # 11
serno      = 3307                |
res3 = 0.0                       | res3 = 9.0
-----
Record key field(s): (Rec. # 12) | Record # 12
serno      = 3308                |
age = 29                          | age = 39
-----
```

While your data file should be correct, there is still a slim chance that it has errors. How is this possible? If by chance the same error was entered in both files (which can happen particularly if the same person enters the data in both records), you will not be able to identify the error. For uniformity, you should overwrite your existing file with the A_EX05F.REC file that is provided with the solution.

Exercise 6: Data safety

At the end of this exercise you should be able to:

- a. Backup and encrypt a data file on an external medium.

Nothing should be more valuable to you than the safety of your data. Unfortunately, people often learn that they should have backed up their data only once they have made the painful experience of having lost them.

EpiData Entry can be very helpful and forcing you to back up your data when you close a file. Whether you back up your data to your hard drive (not a good idea in case of hard drive failure) or an external medium, an additional concern is data security. Let's assume you back up your data to a USB flash memory stick and subsequently you lose it. Anybody who finds your stick can look at your data. Perhaps your data contain confidential information that should not be readable by anybody who is not authorized.

EpiData Entry offers password-only accessible encryption tools using so-called strong encryption that cannot be broken without knowing the password you used for encrypting them. We will make a simple EpiData Entry triplet to show how to back up your data with and without encryption to an external medium.

We will use the following simple dataset of the EpiData promoters:

Name	First Name	Country
Rieder	Hans L	Switzerland
Chiang	Chen-Yuan	Taiwan
Hinderaker	Sven Gudmund	Norway
Katamba	Achilles	Uganda
Tun	Zaw Myo	Myanmar
Nguyen	Binh Hoa	Vietnam
Zwahlen	Marcel	Switzerland

Our A_EX06.QES will look as follows:

Questionnaire for back-up data

```
name          <AAAAAAAAAAAAAAAA>
firstname     <AAAAAAAAAAAAAAAA>
country       <AAAAAAAAAAAAAAAA>
```

Make a REC file and a Check file. In the A_EX06.CHK file make all fields MUSTENTER fields, and then open the Check file which should look as follows:

```
name
  MUSTENTER
END
```

```
firstname
  MUSTENTER
END
```

```
country
  MUSTENTER
END
```

Backing up and encrypting your file

Let us assume that your external back-up device takes drive E and has a folder BACKUP on the drive. The command we add in the check file is then:

```
name
  MUSTENTER
END
```

```
firstname
  MUSTENTER
END
```

```
country
  MUSTENTER
END
```

AFTER FILE

```
  BACKUP e:\foldername ENCRYPT filename mypassword TODAY
END
```

The E:\FOLDERNAME is the folder on the drive of your memory stick which you must have created beforehand. FILENAME is the name of the encrypted file and should best be what it is now, i.e. A_EX06 and “mypassword” is a personally chosen password. *Note: if you forget your password, there will be no way to ever open your backup file.* Of course, you can always see it in your original Check file if you choose to keep that. The TODAY function is optional but a good idea to ensure that the current computer date becomes part of the file name. It must be written after the password.

Note: The command BACKUP will back up all files in that folder. Thus, if you wish to backup up only the files A_EX06.* you must first copy those EpiData files that you wish to back up into another empty folder (such as the c:\temp folder after emptying it).

Re-Opening your back-up file

The back-up file will have the name A_EX06.ZKY and will contain not only the REC file but the entire QES-REC-CHK triplet.

In “Tools” you find “Restore archive” and you are prompted to provide the path and name of your archived and encrypted file, to enter your password, and to provide the path where to restore it, with the option to overwrite any existing files.

Task:

- o Create the QES-REC-CHK triplet, edit the Check file, enter the data above and restore the files.*

Solution to Exercise 6: Data safety

Key Point(s):

- You should always backup data files to avoid loss of important information when the hard drive fails.
- Data files with confidential information saved on an external medium should always be encrypted so that in case you lose it, nobody will be able to read the information.

Task:

- o *Create the QES-REC-CHK triplet, edit the Check file, enter the data above and restore the files.*

Solution:

This is the A_EX06.QES file:

Questionnaire for back-up data

```
name          <AAAAAAAAAAAAAAAA>
firstname     <AAAAAAAAAAAAAAAA>
country       <AAAAAAAAAAAAAAAA>
```

This is the A_EX06.CHK file:

```
name
  MUSTENTER
END
```

```
firstname
  MUSTENTER
END
```

```
country
  MUSTENTER
END
```

```
AFTER FILE
  BACKUP e:\foldername ENCRYPT a_ex06 9suncat TODAY
END
```

Of course, the password used here (“9suncat”) should be replaced by your own.

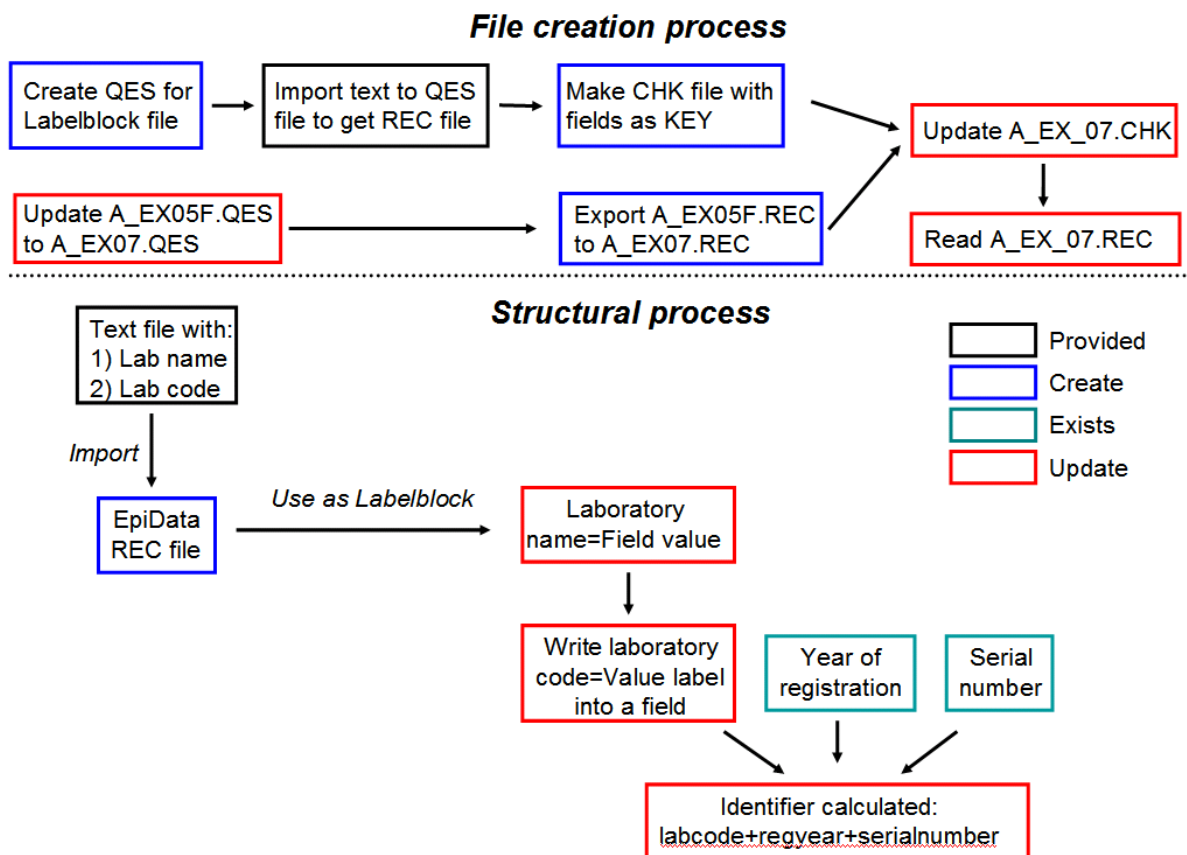
Exercise 7: Using an external file for Labelblocks

At the end of this exercise you should be able to:

- a. Import a text file into EpiData
- b. Use an external file for Labelblocks
- c. Edit and make calculations in the check file
- d. Create a unique identifier from more than one variable

The WHO/Union-recommended Tuberculosis Laboratory Register uses a unique sequential serial number beginning with 1 each year for each examinee. If one were to record data from more than one year, the serial number would no more be unique but a unique identifier could be constructed using a combination of serial number and year of examination. If more than one laboratory is included, then additionally a laboratory code would be needed to unambiguously identify each examinee from several years in several laboratories.

You will accomplish this in this exercise. It will require you to work both in parallel and in sequence. The following graph summarizes the process.



Because the procedure is a bit complex we will explain the sequence of the steps required to arrive at the solution.

Step 1: Make a new QES file and import the supplementary text file to obtain a REC file that will be used as external Labelblock

You need to download the supplementary text file A_EX07_NAMECODE.TXT from the course folder. You must look at this file in your text editor (e.g. with NotePad™ which comes with the Windows operating system. Note: NotePad™ is a rather basic text editor. We recommend using an alternative such as Crimson Editor®, a free and extraordinarily powerful text editor which you may obtain free from <http://www.crimsoneditor.com>). Shown here are just the first 18 lines among the total of 95:

```
1 Awuna;ML_J
2 Beitbridge;MS_D
3 Bindura;MC_A
4 Binga;MW_G
5 Birchenough;ML_M
6 Bonda;ML_I
7 Brunapeg;MS_G
8 Chegutu;MW_J
9 Chikombedzi;MV_I
10 Chimhanda;MC_H
11 Chinhoyi;MW_A
12 Chipinge;ML_B
13 Chiredzi;MV_H
14 Chitamoyo;MW_G
15 Chitsungo;MC_K
16 Chivi;MW_N
17 Chivu;ME_B
18 Collin Saunders;MV_J
```

Note the following here:

- o This is a semicolon-delimited file
- o It has two fields per row, separated by the delimiter
- o The first field is the Name of the laboratory, the second the Laboratory code
- o The Name of the laboratory has various lengths. We can assure you that none has a name exceeding a length of 20
- o The Laboratory code has a field length of 4

This file was created during an operations research project in Zimbabwe (kindly provided by Biggie Mabaera, University of Zimbabwe). At the time of the study, 95 laboratories performed sputum smear microscopy and utilized the standard Tuberculosis Laboratory Register. Each of these laboratories was assigned a unique code of length 4.

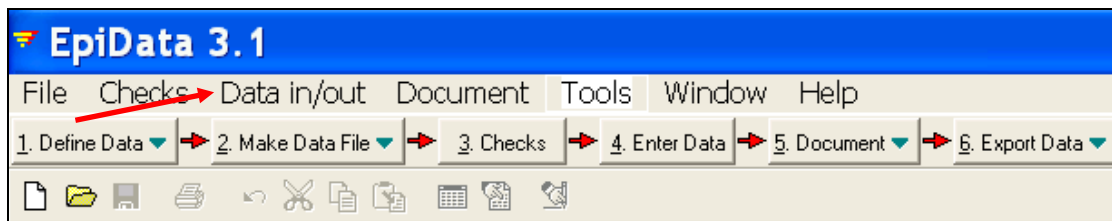
First, a QES file must be prepared into which the two fields will fit. This QES file will be given the name A_EX07_NAMECODE.QES and must have two text fields (*not* Upper case text as it has to accommodate whatever was chosen in the A_EX07_NAMECODE.TXT file). It is a simple file to make:

Questionnaire to accommodate data from the Text file

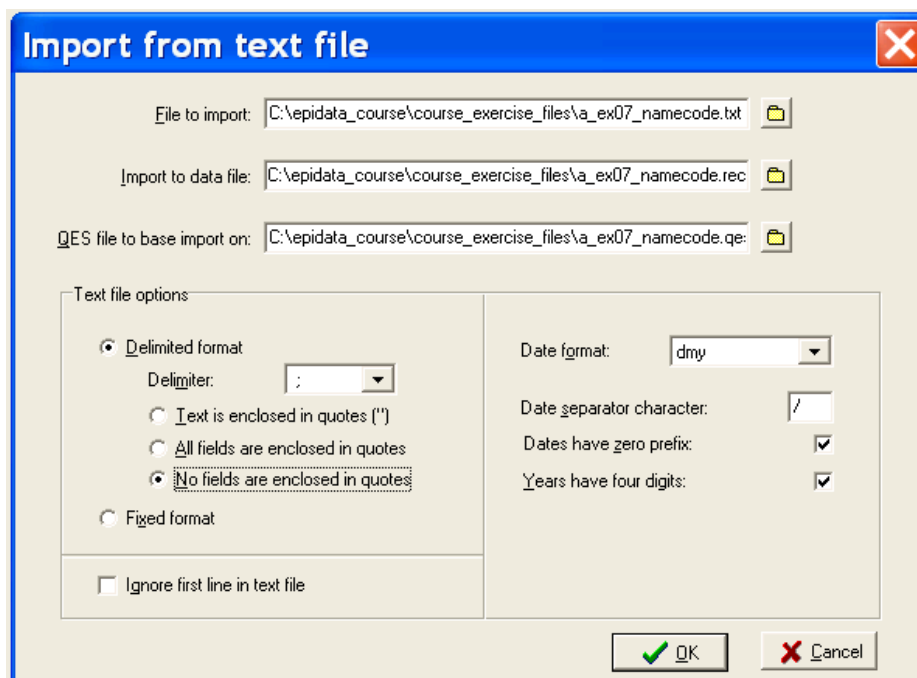
name Name of the laboratory _____
code Code of the laboratory _____

Note: The sequence of the Fields is important. The first field should be the field that will serve as the Field value, the second the one that will serve as the Value label.

No REC file is made as this is created when importing the data from the EpiData Menu Data in/out menu, that is do not go to “2. Make Data File”, but to “Data in/out”:



This allows importing a text file into a pre-existing QES file, resulting in a REC file. Read the menu and options carefully to choose the correct options:

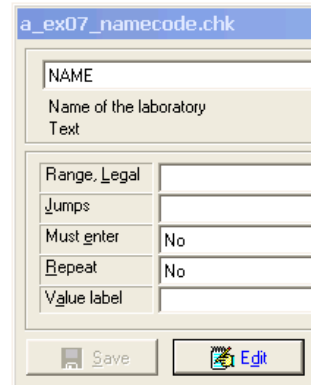


After completing the import, you have to make a CHK file. Use Edit to make each of the fields a KEY field, numbering the keys:

Questionnaire to accommodate data from the Text file

name Name of the laboratory

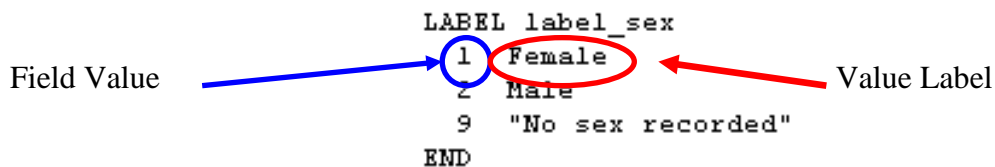
code Code of the laboratory



The field CODE will then be KEY 2.

Role of the KEYS

The command KEY in a field instructs EpiData to create an index for this field (see earlier exercise). This file will be used as a Labelblock, and you remember that a Labelblock has two components the Field Value and the Field Label:



Slightly different, but in analogy, we will now use the field NAME in the A_EX07_NAMECODE.REC file as the Field Value for a field LABNAME in the file A_EX07.REC (see below) and the field CODE as the Value Label for a field LABCODE in that A_EX07.REC file.

When you are done with the CHK file, you have completed this component of the exercise and you have a “A_EX07_NAMECODE” QES-REC-CHK triplet.

Step 2: Making the main QES file A_EX07.QES

To create a unique identifier that consists of three components, use the A_EX05A.QES questionnaire as the starting point, save it as A_EX07.QES and make the necessary amendments.

The unique identifier will have the field name IDCODE and will consist of three components, the code that unambiguously identifies the laboratory (Field length 4), the year of registration (Field length 4), and the laboratory serial number (Field length 4), the components separated by a hyphen (for better visualization of the components only), e.g.:

AA_J-2006-1234

While the calculation for the field IDCODE can obviously be made only after the information on the three fields making it up has been entered, the actual position of the IDCODE in the data entry form is not important. We will arbitrarily place it at the top:

This is the questionnaire for the laboratory register

```

idcode      Laboratory identifier  
labcode     Laboratory code      

labname     Laboratory name      
serno      Laboratory serial number  Write note (F5) if alternate required
regdate    Registration date     Enter 01/01/1800 if not recorded
sex        Examinee's sex      
age        Examinee's age in years  Enter 999 if not recorded
reason     Examination reason   
res1      Result of specimen 1  
res2      Result of specimen 2  
res3      Result of specimen 3  
  
```

Suggestion: it can be useful to place any field that contains information that is calculated by the CHK file and not entered by the data entry person physically separated from the fields which have to be entered (above or below). As the IDCODE will be “calculated” to be the result for 3 fields, it is proposed here to place it at the top.

In previous exercises we had the CHK file writing the Field Value into the field and get the Value Label just typed next to its right during data entry, as for example for the field SEX:

```

sex
  COMMENT LEGAL USE label_sex SHOW
  MUSTENTER
  TYPE COMMENT
END
  
```

resulting during data entry in:

```

sex           Sex of examinee  Female
  
```

The Value label Female does not become part of the REC file (it belongs to the CHK file), only the Field Value 1. As the intention in this exercise is to create an IDCODE that uses the Value label (i.e., the Laboratory code) as one of the three components, this information must be written into a field, and thus be foreseen with a Field length of 4 in the QES file:

This is the questionnaire for the laboratory register

```

idcode      Laboratory identifier  _____
labcode     Laboratory code      _____ ←
labname     Laboratory name      _____
serno      Laboratory serial number #### Write note (F5) if alternate required
regdate    Registration date    <dd/mm/yyyy> Enter 01/01/1800 if not recorded
sex        Examinee's sex      #
age        Examinee's age in years ### Enter 999 if not recorded
reason     Examination reason   #
res1      Result of specimen 1  #.#
res2      Result of specimen 2  #.#
res3      Result of specimen 3  #.#
  
```

With this addition the QES file is complete: it has three new fields: LABNAME that will be entered, and LABCODE that will be the written Value label when the Field value for the new field LABNAME is entered.

Note: After completing the A_EX07.QES file do not make a Data file as this will create an empty data file. Instead, we are going to preserve the data already entered into A_EX05F.REC.

Step 3. Preserving the existing validated dataset

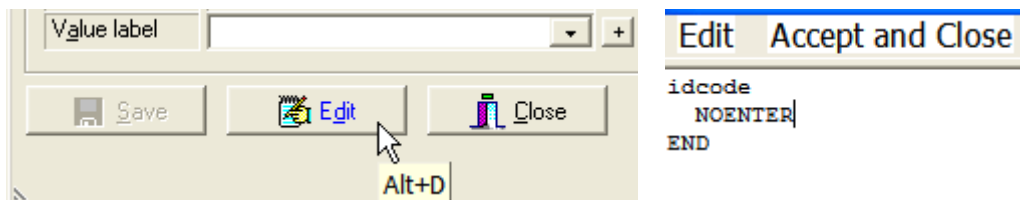
You have already a final, validated A_EX05F.REC file. After you updated the A_EX05F.QES file to the A_EX07.QES file, you must export the A_EX05F.REC file to A_EX07.REC file. This will also give you the A_EX05F.CHK file, copied to a A_EX07.CHK file.

Step 4. Editing the main CHK file A_EX07.CHK

We start by opening the A_EX07.REC file (created by the export) and you will be prompted to update it to the A_EX07.QES file. After affirming, close the file, and start editing the A_EX07.QES file.

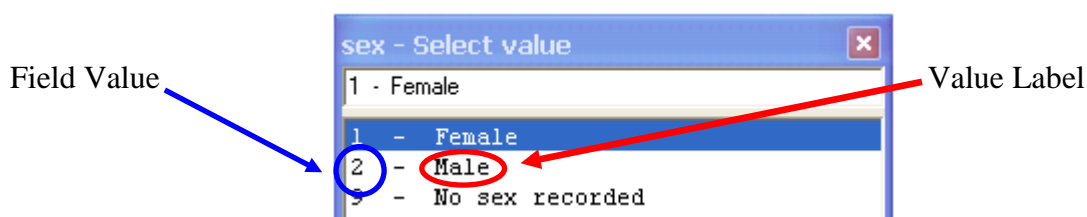
The first step might be editing all three new fields:

idcode
labcode
labname

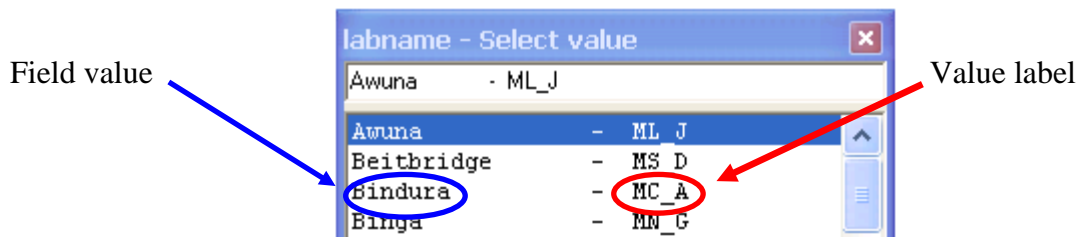


making them NOENTER fields.

When we entered the field SEX during data entry, the Labelblock showed:



Similarly, we want for LABNAME:



The A_EX07_NAMECODE.REC file will be invoked to serve as a Labelblock. Of course, we could make Labelblocks in the CHK file as introduced in an earlier exercise. If the list of values is very large, this becomes tedious work and may make for a very long check file, and at some point may even exceed the legal limits. In such an instance it may become more efficient to use another EpiData REC file that is invoked.

The notation in the example for the field SEX in the CHK file was:

```
sex
  COMMENT LEGAL USE label_sex SHOW
  MUSTENTER
  TYPE COMMENT
END
```

As you learned, the command TYPE COMMENT has the effect that the Value label is written to the right of the field SEX. Alternatively you can tell EpiData Entry to write the Value label into another field:

```
varx
  COMMENT LEGAL USE label_x SHOW
  MUSTENTER
  TYPE COMMENT otherfield
END
```

The grammar to accomplish the same thing with an external REC file rather than a Labelblock is very similar:

```
labname
  MUSTENTER
  COMMENT LEGAL a_ex07_namecode.rec SHOW
  TYPE COMMENT labcode
END
```

Note that USE is dropped from the command.

Once you have the relation made to these external files, you will have the Field value for the field LABCODE which you need for creating the identifier. Creating the identifier is like making a calculation. If you have two numeric fields, where the third field summarizes the information from the two numeric fields, we basically have:

```
field1
* some commands
END

field2
* some commands
```

```

AFTER ENTRY
  field3=field1+field2
END
END

field3
  NOENTER
END

```

If FIELD1 and FIELD2 are numeric fields with values 2 and 6 respectively then FIELD3 will get the value 8. If FIELD1 and FIELD2 are text fields with values “AB” and “XYZ” respectively, then FIELD3 will have the value “ABXYZ”. The identifier can obviously be created only after all the three values required are available, and this is placed in an AFTER ENTRY block:

```

regdate
  RANGE 01/01/2000 31/12/2005
  LEGAL
    01/01/1800
  END
  MUSTENTER
  AFTER ENTRY
  idcode=labcode+"-"+year(regdate)+"-"+serno
  END
END

```

Note above the function to extract the year from a date field.

Finally, we want to prevent a data entry person from ever even getting into the field IDCODE. This is accomplished by changing the MUSTENTER into:

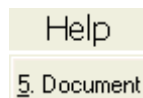
```

idcode
  NOENTER
END

```

Note: Do not make the field that takes the laboratory name (the Field value) a REPEAT field. In the current version of EpiData Entry, the Check commands in a field that uses an external file as a Labelblock will not be properly executed. Here the Value label will not be written into the field made for it and no IDCODE will thus be created.

Finally, before you complete this exercise, we encourage you to try the help file:



which you can access by clicking of with the shortcut **Alt+H**. **Please use it, that’s what it is for.**

Important note: Although later IDCODE will become the unique identifier, for the time being you must keep SERNO as the unique identifier. Why is that so? If you were to make IDCODE as the unique identifier, you could not update your A_EX07.REC file, because all 15 records are missing it (thus non-unique).

Step 5. Updating the A_EX07.REC file

Open now the A_EX07.REC file, go to the first record, add the LABNAME field, and keep pressing Enter to automatically update the three new fields:

IDCODE
LABCODE
LABNAME

Then go to the next record, and so on, until all 15 records are updated.

Step 6. Finalize the A_EX07.CHK file

As the last step, change the A_EX07.CHK file to make IDCODE a KEY UNIQUE field and remove that instruction from the SERNO field.

Tasks:

- o Make the QES file A_EX07_NAMECODE.QES*
- o Import the text file A_EX07_NAMECODE.TXT which creates the A_EX07_NAMECODE.REC file*
- o Make the two fields in the Check file A_EX07_NAMECODE.CHK Key fields*
- o Edit the A_EX07.QES file*
- o Export the A_EX05F.REC file to A_EX07.REC*
- o Open the A_EX07.REC file to adjust its structure from the newer A_EX07.QES file, then close it*
- o Edit the A_EX07.CHK file to:
 Use the file A_EX07_NAMECODE.REC instead of a Labelblock for the Fields identifying the laboratory
 Create an identifier IDCODE from the three fields LABCODE, the year of REGDATE, and SERNO*
- o Update the A_EX07.REC file with LABNAME (and the three fields calculated) for all 15 records*
- o Finalize the A_EX07.CHK file to make IDCODE instead of SERNO the KEY UNIQUE*

Solution to Exercise 7: Using an external file for Labelblocks

Key Point(s):

- An external file for Labelblocks is more efficient when the list of values is very large and would result in a very long check file.

Tasks:

- o *Make the QES file A_EX07_NAMECODE.QES*
- o *Import the text file A_EX07_NAMECODE.TXT which creates the A_EX07_NAMECODE.REC file*
- o *Make the two fields in the Check file A_EX07_NAMECODE.CHK Key fields*
- o *Edit the A_EX07.QES file*
- o *Export the A_EX05F.REC file to A_EX07.REC*
- o *Open the A_EX07.REC file to adjust its structure from the newer A_EX07.QES file, then close it*
- o *Edit the A_EX07.CHK file to:*
 - Use the file A_EX07_NAMECODE.REC instead of a Labelblock for the Fields identifying the laboratory*
 - Create an identifier IDCODE from the three fields LABCODE, the year of REGDATE, and SERNO*
- o *Update the A_EX07.REC file with LABNAME (and the three fields calculated) for all 15 records*
- o *Finalize the A_EX07.CHK file to make IDCODE instead of SERNO the KEY UNIQUE*

Solution

The A_EX07_NAMECODE.QES file:

```
name _____
code _____
```

The A_EX07_NAMECODE.CHK file:

```
name
  MUSTENTER
  key 1
END
```

```
code
  MUSTENTER
```

key 2
END

The A_EX07.QES file:

This is the questionnaire for the laboratory register

idcode	Laboratory identifier	<input type="text"/>	
labcode	Laboratory code	<input type="text"/>	
labname	Laboratory name	<input type="text"/>	
serno	Laboratory serial number	<input type="text"/>	Write note (F5) if alternate required
regdate	Registration date	<input type="text"/>	Enter 01/01/1800 if not recorded
sex	Examinee's sex	<input type="text"/>	
age	Examinee's age in years	<input type="text"/>	Enter 999 if not recorded
reason	Examination reason	<input type="text"/>	
res1	Result of specimen 1	<input type="text"/>	
res2	Result of specimen 2	<input type="text"/>	
res3	Result of specimen 3	<input type="text"/>	

The A_EX07.CHK file:

```
LABELBLOCK
  LABEL label_sex
    1 Female
    2 Male
    9 "No sex recorded"
  END
  LABEL label_reason
    0 Diagnosis
    8 "Follow-up, month not stated"
    9 "Reason not stated"
    1 "Follow-up at 1 month"
    2 "Follow-up at 2 months"
    3 "Follow-up at 3 months"
    4 "Follow-up at 4 months"
    5 "Follow-up at 5 months"
    6 "Follow-up at 6 months"
    7 "Follow-up at 7 months or later"
  END
  LABEL label_result
    0.0 Negative
    1.0 "1+ positive"
    2.0 "2+ positive"
    3.0 "3+ positive"
    4.0 "4+ positive"
    9.0 "No result recorded"
    5.0 "Positive, not quantified"
    6.0 "Scanty, not quantified"
    0.1 "Scanty, 1 AFB per 100 fields"
    0.2 "Scanty, 2 AFB per 100 fields"
    0.3 "Scanty, 3 AFB per 100 fields"
```

```

    0.4 "Scanty, 4 AFB per 100 fields"
    0.5 "Scanty, 5 AFB per 100 fields"
    0.6 "Scanty, 6 AFB per 100 fields"
    0.7 "Scanty, 7 AFB per 100 fields"
    0.8 "Scanty, 8 AFB per 100 fields"
    0.9 "Scanty, 9 AFB per 100 fields"
END
END

labcode
  NOENTER
END

idcode
  NOENTER
  KEY UNIQUE
END

labname
  COMMENT LEGAL a_ex07_namecode.rec SHOW
  MUSTENTER
  TYPE COMMENT labcode
END

serno
  MUSTENTER
END

regdate
  RANGE 01/01/2000 31/12/2005
  LEGAL
    01/01/1800
  END
  MUSTENTER
  AFTER ENTRY
    idcode=labcode+"-"+year(regdate)+"-"+serno
  END
END

sex
  COMMENT LEGAL USE label_sex SHOW
  MUSTENTER
  TYPE COMMENT
END

age
  RANGE 0 125
  LEGAL
    999
  END
  MUSTENTER
END

reason
  COMMENT LEGAL USE label_reason SHOW
  MUSTENTER
  TYPE COMMENT
END

res1
  COMMENT LEGAL USE label_result SHOW

```

```
MUSTENTER
TYPE COMMENT
END
```

```
res2
COMMENT LEGAL USE label_result SHOW
MUSTENTER
TYPE COMMENT
END
```

```
res3
COMMENT LEGAL USE label_result SHOW
MUSTENTER
TYPE COMMENT
END
```

```
AFTER RECORD
IF idcode=. THEN
HELP "Core information missing:\n LABNAME, SERNO, and REGDATE\n must
all be available" TYPE=WARNING
GOTO labname
ENDIF
END
```

Exercise 8: Dealing with incomplete dates

At the end of this exercise you should be able to:

- a. Approximate dates when day and/or month is missing
- b. Create a date from three component fields (day, month, year)
- c. Make use of temporary variables in the check file to make calculations.

Date information is commonly collected, most commonly perhaps for the calculation of intervals or another purpose. However, information on the date is often incomplete. For instance, if a date of symptom onset is asked from a patient, the patient may remember the year only, or the year and month only. The date of onset is thus unknown, and an interval between the current doctor's visit (which might be known exactly) and date of symptom onset cannot be calculated and must remain unknown. Nevertheless, if for instance year and month of symptom onset are known (but not the day) and the date of the actual visit is known exactly, something is known, and an approximate interval could be provided. This exercise will offer a way to make such approximations.

As we should preferably not leave it up to the data entry person to enter incorrect, approximated dates, we must avoid requiring to enter a date, but rather split it into its components year, month, and day, and then reassemble it.

Let us assume that we wish to calculate the variables:

```
regexct      Exact registration date <dd/mm/yyyy>
regappr      Approximate registration date <dd/mm/yyyy>
```

from the fields:

```
regdd        Day of registration ##
regmm        Month of registration ##
regyy        Year of registration #####
```

The field REGEXCT will be an existing date within the legal range currently defined if all three components of the date are known else its value will be set to 01/01/1800. The Field REGAPPR will be equal to REGEXCT if the latter is within the legal range. If the day only is unknown, the day value for REGAPPR will be 15 (mid-month), if both month and day are unknown, the month will be 07 and the day 01 (mid-year). If all three components are unknown, the value of REGAPPR will be set to 01/01/1800.

We will impose a hierarchical structure, i.e., that if a value for an unknown month is entered, a known day – if entered as such – thus does not make sense, and if the year is unknown, neither a known month or known day will be affect the calculation.

Thus, we will replace the current field REGDATE with the above five fields, three of which (REGDD, REGMM, and REGYY) are entered while the other two (REGEXCT and REGAPPR) are calculated from the former three (and will thus be NOENTER Fields).

In addition, we will add another variable that measures the quality of the calculated date information, and that we might name

```
regqual      Quality of registration date #
```

This field can take the values 0, 1, 2, and 3, where 0 (zero) indicates that none is known, and 3 that all three date components are known.

For this task, we introduce temporary variables. Although for this specific example, it would be possible to solve it without such variables, it seems as good as any occasion to introduce this possibility for this exercises. Temporary variables can have a field length of 16 and they do not appear in the QES file and will not become part of the dataset. They will be used in the Check file for internal calculations in a BEFORE FILE block as in:

```
BEFORE FILE
  DEFINE regddTemp ##
  DEFINE regmmTemp ##
  DEFINE regyyTemp ####
END
```

BEFORE FILE means exactly what it says: Before anything is entered into the file, these variables are created. There is also a Check file command BEFORE RECORD (look up in the Help file what these commands do).

To create a date from the three component fields, the Check file command is:

```
fulldate=DATE(varday,varmonth,varyear)
```

Tasks:

- o Create an A_EX08.* triplet (using the A_EX07.* files as the starting point). The questionnaire should display all the calculated variables.*
- o Create the A_EX08.REC*
- o Edit the A_EX08.CHK file to make the calculations. Note that you will need to define temporary variables for this task.*
- o Enter some data to check the functionality*

Solution to Exercise 8: Dealing with incomplete dates

Key Point(s):

- Approximating dates in a systematic way will enable you to approximate intervals e.g. between date of symptom onset and visit to the health facility.
- Do not leave it to the data entry person to enter incorrect or approximate dates.

Tasks:

- o Create an A_EX08.* triplet (using the A_EX07.* files as the starting point). The questionnaire should display all the calculated variables.
- o Create the A_EX08.REC
- o Edit the A_EX08.CHK file to make the calculations. Note that you will need to define temporary variables for this task.
- o Enter some data to check the functionality

Solution

The A_EX08.QES file:

This is the questionnaire for the laboratory register

labcode	Laboratory code	<input type="text"/>	
idcode	Laboratory identifier	<input type="text"/>	
regexct	Exact registration date	<input type="text"/>	Set to 01/01/1800 if any unknown
regappr	Approximate registration date	<input type="text"/>	Set to 01/01/1800 if year unknown
regqual	Quality of registration date	<input type="text"/>	
labname	Laboratory name	<input type="text"/>	
regdd	Day of registration	<input type="text"/>	Enter 99 if not recorded
regmm	Month of registration	<input type="text"/>	Enter 99 if not recorded
regyy	Year of registration	<input type="text"/>	Enter 9999 if not recorded
serno	Laboratory serial number	<input type="text"/>	Assign 9001,9002,... if not unique (write note (F5))
sex	Examinee's sex	<input type="text"/>	
age	Examinee's age in years	<input type="text"/>	Enter 999 if not recorded
reason	Examination reason	<input type="text"/>	
res1	Result of specimen 1	<input type="text"/>	
res2	Result of specimen 2	<input type="text"/>	
res3	Result of specimen 3	<input type="text"/>	

The A_EX08.CHK file (pertinent parts only):

```
LABELBLOCK
  LABEL label_sex
    1 Female
    2 Male
```

```

    9 "No sex recorded"
    END
...
...
END

BEFORE FILE
    DEFINE regddTemp ##
    DEFINE regmmTemp ##
    DEFINE regyyTemp ####
END

idcode
    NOENTER
    KEY UNIQUE 1
END

regexct
    NOENTER
END

regappr
    NOENTER
END

regqual
    NOENTER
END

labname
    COMMENT LEGAL a_ex07_namecode.rec SHOW
    MUSTENTER
    TYPE COMMENT labcode
END

labcode
    NOENTER
END

regdd
    RANGE 1 31
    LEGAL
        99
    END
    MUSTENTER
END

regmm
    RANGE 1 12
    LEGAL
        99
    END
    MUSTENTER
    REPEAT
END

regyy
    RANGE 2000 2004
    LEGAL
        9999
    END
    MUSTENTER
    REPEAT
    AFTER ENTRY
        regddTemp=regdd
        regmmTemp=regmm
        regyyTemp=regyy
    IF (regdd=99) or (regmm=99) or (regyy=9999) THEN

```

```

    regexct="01/01/1800"
ELSE
    regexct=date(regddTemp,regmmTemp,regyyTemp)
    regappr=regexct
    regqual=3
ENDIF
IF regdd=99 THEN
    regddtemp=15
    regqual=2
ENDIF
IF regmm=99 THEN
    regddTemp=01
    regmmTemp=07
    regqual=1
ENDIF
IF regyy=9999 THEN
    regddTemp=01
    regmmTemp=01
    regyyTemp=1800
    regqual=0
ENDIF
    regappr=date(regddTemp,regmmTemp,regyyTemp)
END
END

serno
    MUSTENTER
    AFTER ENTRY
        idcode=labcode+"-"+regyyTemp+"-"+serno
    END
END

```

...

A completed A_EX08.REC record:

This is the questionnaire for the laboratory register

labcode	Laboratory code	<input type="text" value="MV_I"/>	
idcode	Laboratory identifier	<input type="text" value="MV_I-2003-2341"/>	
regexct	Exact registration date	<input type="text" value="01/01/1800"/>	Set to 01/01/1800 if any unknown
regappr	Approximate registration date	<input type="text" value="15/11/2003"/>	Set to 01/01/1800 if year unknown
regqual	Quality of registration date	<input type="text" value="2"/>	

labname	Laboratory name	<input type="text" value="Chikombedzi"/>	
regdd	Day of registration	<input type="text" value="99"/>	Enter 99 if not recorded
regmm	Month of registration	<input type="text" value="11"/>	Enter 99 if not recorded
regyy	Year of registration	<input type="text" value="2003"/>	Enter 9999 if not recorded
serno	Laboratory serial number	<input type="text" value="2341"/>	Assign 9001,9002,... if not unique (write note (F5))
sex	Examinee's sex	<input type="text" value="1"/>	Female
age	Examinee's age in years	<input type="text" value="24"/>	Enter 999 if not recorded
reason	Examination reason	<input type="text" value="0"/>	Diagnosis
res1	Result of specimen 1	<input type="text" value="0.0"/>	Negative
res2	Result of specimen 2	<input type="text" value="1.0"/>	1+ positive
res3	Result of specimen 3	<input type="text" value="0.7"/>	Scanty, 7 AFB per 100 fields

Exercise 9: Keeping track of data entry time

At the end of this exercise you should be able to:

- a. Edit the check file, and use temporary variables to calculate the amount of time required for data entry.

In this exercise you will learn to calculate the number of seconds which are required to complete one record. A field to retain this value as part of the database thus needs to be added to the questionnaire.

If you prepare a study, you should make it a rule to enter several hundred records by yourself (with the help of a colleague). This is important for two reasons:

- o Identify weaknesses in the data entry form and the CHK file
- o Recording the time needed to enter the information

You need the information on recording time to know what you can require from another data entry person. This is critical for making your budget. It is obviously not satisfactory to pay a data entry person according to the time the person claims to need: some people are slow and they should not receive the same remuneration as the faster person. If you take your achievement as the basis, you can objectively pay for the time required if a data entry person works at your speed: those slower than you will lose, those faster will win, and that is how it should be.

We can use EpiData Entry to monitor our data entry time and make a permanently available record that we can later analyze. In the EpiData\samples\ folder you find three files:

```
DateTime.qes  
DateTime.rec  
DateTime.chk
```

These three files are the basis to adjust our own sample files together with the **About time** in the EpiData Entry Help file. It is not an easy task and while you would certainly be able to figure it out by yourself, we will give you a helping hand.

We are going to make use of the clock of the computer. EpiData Entry has a function NOW which records the exact time of the computer. This time consists of the date and the time of the day. NOW writes the information on the date into the integer part of a field and the time of the day as a fraction of the 24-hour clock into the fractional part of the field. If we define thus such a field for the file and start counting the time when a new record is opened:

```
BEFORE FILE  
  DEFINE StartTime #####.#####  
END  
  
BEFORE RECORD  
  StartTime=NOW  
END
```

We could of course write the field `StartTime` into the questionnaire as a `NOENTER` field and then we would get, as an example:

39650.864672

NOW gives the computer time right at this point, writing the date as the integer part and the fraction of the day as the fractional part. The computer stores the date as the number of days passed since the internal anchor date. In EpiData, this anchor date is 31 December 1899. If we divide the above integer part of the number by the average number of days per year we get $39650/365.25=108.56$, that is a bit more than 108.5 years after the anchor date, or some time in July 2008. What about the fractional part? If we multiply the 24-hour day by this fraction we get $0.864672*24=20.75$ which is roughly a quarter to nine in the evening, but it is much more precise than to the quarter (6 digits!) because what we really want is to have it expressed in seconds and one day has 86,400 seconds. To get the rounding proper we use even 6 rather than the bare minimum of 5 digits.

As the database only needs to retain the number of seconds required to complete one record, the number above is thus just something we need the CHK file to calculate in the background.

Assuming that we have a field SECONDS in the data file, we would then write after entering the value for the last field:

```
res3
  COMMENT LEGAL USE label_result SHOW
  MUSTENTER
  TYPE COMMENT
  AFTER ENTRY
    IF seconds=. THEN
      seconds=(NOW-StartTime)*86400
    ENDIF
  END
END

AFTER RECORD
  IF idcode=. THEN
    HELP "Core information missing:\n      LABNAME, SERNO, and REGDATE\n must
all be available" TYPE=WARNING
    GOTO labname
  ENDIF
END
```

This NOW is a different NOW from the beginning (computer clock!) and all we do is to revert the difference between the two time points into seconds.

If we do it as above, the clock will stop right after the value of the last field RES1 has been entered and that value (number of seconds) will be written into the field SECONDS. Even when returning to the record, the original value in the field SECONDS will not change. If we were to take out the:

```
IF seconds=. THEN
  seconds=(NOW-StartTime)*86400
ENDIF
```

then the number of seconds would change to a new value if going again later through the record, and the value then might be lower. Ideally, however, the number of SECONDS would be cumulative, i.e. if a record is re-visited, the additional time during the revisit would be added to the pre-existing time. In other words, the time for corrections after validation would

be added. The change required to accomplish the latter is to delete it from the AFTER ENTRY statement in the RES1 field:

```
res3
  COMMENT LEGAL USE label_result SHOW
  MUSTENTER
  TYPE COMMENT
  AFTER ENTRY
  IF seconds=. THEN
    seconds=(NOW-StartTime)*86400
  ENDIF
END
END
```

Instead, the statements are integrated into the AFTER RECORD statement and extended to provide accumulation of time:

```
AFTER RECORD
  IF idcode=. THEN
    HELP "Core information missing:\n      LABNAME, SERNO, and REGDATE\n must
all be available" TYPE=WARNING
    GOTO labname
  ENDIF
  IF seconds=. THEN
    seconds=(NOW-StartTime)*86400
  ELSE
    seconds=seconds+(NOW-StartTime)*86400
  ENDIF
END
```

However, an even more informative approach is to have both the information about time required to enter one record for the first time and the information about the total amount of time (cumulatively) spent on a records, i.e. including the time spent on corrections. In the analysis one can then thus determine how much time is spent on entering one record, and how much additional time on correcting the record if a discordance must be resolved after validation (by subtracting the value in the field for first entry from that of the cumulative time). This is important because it has been shown that working faster is accompanied by making more errors, thus requiring revisiting more records again.

Note on formatting: The choice of a mixture of lower-case and upper-case letters (eg, in StartTime) is for easier visual recognition only. As mentioned earlier, EpiData Entry is not case-sensitive. This makes it particularly powerful as you can make use of formatting to visual ease. As a general rule, if you let EpiData Entry make the choices for you, commands are capitalized, all the rest is not. As for field names, lower-case is always preferred. While it does not matter in formatting, you can force its format in the REC file to any option you prefer, but we give preference to lower case (go to "File" "Options" "Create data file").

Task:

- o Start with the A_EX08 QES-REC-CHK files, save them as A_EX09 QES-REC-CHK and revise them accordingly. Don't just retype the above, try to consider the logic of it!*

- o Make two NOENTER fields, one that calculates the data entry time for the first entry (that will not change by revisiting the records) and another field that calculates the cumulative time resulting from one or more re-visits of the record*
- o Enter some data to check the functionality.*

Solution to Exercise 9: Keeping track of data entry time

Key Point(s)

- The amount of time required to enter a record is critical for making a budget, and payment can be objectively made.

Task:

- o Start with the *A_EX08 QES-REC-CHK* files, save them as *A_EX09 QES-REC-CHK* and revise them accordingly. Don't just retype the above, try to consider the logic of it!
- o Make two *NOENTER* fields, one that calculates the data entry time for the first entry (that will not change by revisiting the records) and another field that calculates the cumulative time resulting from one or more re-visits of the record
- o Enter some data to check the functionality.

Solution

The *A_EX09.QES* file:

This is the questionnaire for the laboratory register

labcode	Laboratory code	<input type="text"/>	
idcode	Laboratory identifier	<input type="text"/>	
regexct	Exact registration date	<input type="text"/>	Set to 01/01/1800 if any unknown
regappr	Approximate registration date	<input type="text"/>	Set to 01/01/1800 if year unknown
regqual	Quality of registration date	<input type="text"/>	
seconds	Number of seconds for record	<input type="text"/>	
cumsecs	Cumulative number of seconds	<input type="text"/>	
labname	Laboratory name	<input type="text"/>	
regdd	Day of registration	<input type="text"/>	Enter 99 if not recorded
regmm	Month of registration	<input type="text"/>	Enter 99 if not recorded
regyy	Year of registration	<input type="text"/>	Enter 9999 if not recorded
serno	Laboratory serial number	<input type="text"/>	Assign 9001,9002,... if not unique (write note (F5))
sex	Examinee's sex	<input type="text"/>	
age	Examinee's age in years	<input type="text"/>	Enter 999 if not recorded
reason	Examination reason	<input type="text"/>	
res1	Result of specimen 1	<input type="text"/>	
res2	Result of specimen 2	<input type="text"/>	
res3	Result of specimen 3	<input type="text"/>	

The *A_EX09.CHK* file (only the pertinent parts):

...

```

BEFORE FILE
  DEFINE regddTemp ##
  DEFINE regmmTemp ##
  DEFINE regyyTemp ####
  DEFINE StartTime #####.#####
END

BEFORE RECORD
  StartTime=NOW
END

AFTER RECORD
  IF idcode=. THEN
    HELP "Core information missing:\n      LABNAME, SERNO, and REGDATE\n must all be
available" TYPE=WARNING
    GOTO labname
  ENDIF
  IF seconds=. THEN
    seconds=(NOW-StartTime)*86400
  ENDIF
  IF cumsecs=. THEN
    cumsecs=(NOW-StartTime)*86400
  ELSE
    cumsecs=cumsecs+(NOW-StartTime)*86400
  ENDIF
END

labcode
  NOENTER
END

```

...

A completed record

This is the questionnaire for the laboratory register

labcode	Laboratory code	<input type="text" value="ML_J"/>	
idcode	Laboratory identifier	<input type="text" value="ML_J-2004-1234"/>	
regexct	Exact registration date	<input type="text" value="12/06/2004"/>	Set to 01/01/1800 if any unknown
regappr	Approximate registration date	<input type="text" value="12/06/2004"/>	Set to 01/01/1800 if year unknown
regqual	Quality of registration date	<input type="text" value="3"/>	
seconds	Number of seconds for record	<input type="text" value="43"/>	
cumsecs	Cumulative number of seconds	<input type="text" value="58"/>	
labname	Laboratory name	<input type="text" value="Awuna"/>	
regdd	Day of registration	<input type="text" value="12"/>	Enter 99 if not recorded
regmm	Month of registration	<input type="text" value="6"/>	Enter 99 if not recorded
regyy	Year of registration	<input type="text" value="2004"/>	Enter 9999 if not recorded
serno	Laboratory serial number	<input type="text" value="1234"/>	Assign 9001,9002,... if not unique (write note (F5))
sex	Examinee's sex	<input type="text" value="2"/>	
age	Examinee's age in years	<input type="text" value="22"/>	Enter 999 if not recorded
reason	Examination reason	<input type="text" value="0"/>	
res1	Result of specimen 1	<input type="text" value="1.0"/>	
res2	Result of specimen 2	<input type="text" value="0.7"/>	
res3	Result of specimen 3	<input type="text" value="9.0"/>	