Exercise 1:

side 71-71 i DB-chap01

1.6 Theme:

Ejer og kældedyr

1.8 primary key

OwnerLastName OwnerFirstName OwnerPhone OwnerEmail

PetId PetName PetType PetBreed PetDOB

1.9 foreign key (henviser til en primary key i en anden tabel (skaber sammenspil mellem to primary keys))

Id OwnerPhone (FK) PetId (FK) 1.12 foreign key Id OwnerPhone (FK) PetId (FK) Service Date Charge

1.13 Vi sørger for at der er konsistens fordi det er en foreign key, og hvis du sletter noget ændres det også i hele systemet.

Exerciese 2: done

Exercise 3: side 72 i DB-chap01

1.18

* Create and process forms
* Process user queries
* Create and process reports
* Execute application logic
* Control applications

1.19

Metadata: Information om det tabellerne der er i databasen. En relationel database skal indeholde selvbeskrivende tabel navne og kol(l)on(n)e navne. Metadata kan være tabel navne, database navne, kolonne navne, egenskaber af tabeller og kolonner.

1.22 Formålet med en DBMS:

* Create database
* Create tables
* Create supporting structures ( e.g. indexes)
* Read database data
* Modify (insert, update or delete) Database data
* Maintain database structures
* Enforce rules
* Control concurrency
* Provide security
* Perform backup and recovery

Man bruger den til at oprette, administrere databaser.

1.23 Kig på liste i 1.22

Exerceise 4: done

Exerciese 5:

side 72-73 DB-chap01

1.31

A: Themes:

Item(number, describtion)

Acquisition(number(FK), acDate og acPrize)

B: Themes: IndustryType(IndustryId(PR), IndustryCode)

Company(CompanyId(PK), Name, IndustryId(FK), GrossSales)

Office(OfficeName, OfficeTitle, CompanyId(FK))

C: Themes:

IndustryType(IndustryId, IndustryCode)

Company(CompanyId(PR),Name, IndustryId(FK), GrossSales, NameOfPress)

D: Themes:

Harddisk(Model(PR),DiscType, DiscCapasity)

Computer(SerialNumber(PR), Make, Model(FK))

E: Themes:

Person(PersonId(PR), Name, DateOfHire, DepartmentId(FK))

Department(Name, DepartmentManager, DepartmentId(PK)) Project(ProjectId, Name, ProjectManager, DetpartmentId(FK))

ProjectGroup(ProjectId(FK), PersonId(FK), ProjectGroupId(PR))

1.32 Hvilke problemer er der i de opdaterede A,B,C,D,E:

A: Du kan ikke lave acquisition uden at du har oprettet et item først, fordi du skal bruge ItemId for at lave en acquisition. Hvis du sletter ItemId, kan du ikke lave en acquisition.

B: Hvis du sletter IndustryId eller CompanyId, kommer deres foreign key til at blive messed up. Hvis vi ændrer Name, kommer Industry og Company ikke til at blive opdateret. Ændring af foreign key skaber også inkonsistens. Tilføjelse af foreign key, uden at en tilsvarende priamry key eksistere.

C: Same shit

D: Same shit

E: Same shit

Exercise 6: Done

Exercise 7: side 75-76 DB-chap01

Lavet sammen med Faizan, Andreas, Lars og Mikkel.

San Juan Opgave A

Owner (OwnerName, OwnerPhone (PK), BillingAdress )

Boat (BoatId (PK), BoatName, Make, Model, OwnerPhone (FK), BoatLength)

Opgave B

Hvis man sletter en linje kan man miste information om en ejer eller information om en båd.

Hvis den samme Ejer står for flere både og man kun opdaterer en af rækkerne, er der pludselig inkonsistens mellem dataerne i resten af listen.

Ved tilføjelse af en ny båd skal man have information om ejeren da de to typer af data hører sammen.

Opgave C

De to tabeller er stadig afhængige af hinanden.

Opgave D

Charter (CharterID (PK), CharterDate, CharterCustomerId (FK), ChargedAmount, BoatId (FK) )

Customer ( CustomerId , CustomerName…)

Opgave E

Stor mængde redundant data

Der går for meget data tabt ved sletning

Ændring skaber inkonsistens

Når man tilføjer mangler data.

Opgave F

Owner (OwnerName, OwnerPhone (PK), BillingAdress )

Boat (BoatId (PK), BoatName, Make, Model, OwnerPhone (FK), BoatLength)

Charter (CharterID (PK), CharterDate, CharterCustomerId (FK), ChargedAmount, BoatId (FK) )

Customer ( CustomerId , CustomerName…)

Tabellerne er indbyrtes afhængige.

CRUD (Create , Read, Update, Delete)