Sample Relational Database

Table of Contents

1 Database Schema: ................................................................................................ 2
2 Functional Dependencies: .................................................................................... 3
3 A current state of the database: ........................................................................ 4
4 Database Queries (relational Algebra): .............................................................. 5
5 Database Queries (relational Calculus): ............................................................ 6
6 Database Queries (SQL): ..................................................................................... 7
7 Practical implementation of the database with mySQL ................................... 8
1 Database Schema:

- **Domains:**
  - FNr Integer; factory number
  - FName String; factory name
  - FStreet String; factory address part (street)
  - FZip String; factory address part (zip code)
  - FCity String; factory address part (city)
  - ENr Integer; employee number
  - EName String; employee name
  - EPosition String; employee position
  - Qnt Integer; quantity of a type
  - TNr Integer; type number
  - TIdentifier String; type identifier
  - TData String; type data
  - TrNr Integer; transfer number
  - Date Date; date of the transfer
  - SrcF Integer; source factory
  - DestF Integer; destination factory
  - Complete Integer; binary value indicating the completion of the Transfer

- **Relation:** Factory (FNr, FName, FStreet, FZip, FCity)
- **Relation:** Employee (ENr, FNr, EName, EPosition)
- **Relation:** Vehicular Inventory (FNr, TNr, Qnt)
- **Relation:** Type (TNr, TIdentifier, TData)
- **Relation:** Transfer (TrNr, ENr, Date, SrcF, DestF, TNr, Qnt, Complete)
2 Functional Dependencies:

- Relation: Factory (FNr, FName, FStreet, FZip, FCity)
  - FNr → FName
  - FNr → FStreet
  - FNr → FZip
  - FNr → FCity

- Relation: Employee (ENr, FNr, EName, EPosition)
  - ENr → FNr
  - ENr → EName
  - ENr → EPosition

- Relation: Vehicular_Inventory (FNr, TNr, Qnt)
  - (FNr, TNr) → Qnt

- Relation: Type (TNr, TIdentifier, TData)
  - TNr → TIdentifier
  - TNr → TData

- Relation: Transfer (TrNr, ENr, Date, SrcF, DestF, TNr, Qnt, Complete)
  - TrNr → ENr
  - TrNr → Date
  - TrNr → SrcF
  - TrNr → DestF
  - TrNr → TNr
  - TrNr → Qnt
  - TrNr → Complete

A relation is in 3rd Normal Form if Every data item outside the primary key is identifiable

- by the primary key
- by the whole primary key
- only by the primary key

Referring to the section “Functional Dependencies” it is quite obvious, that all relations of the database are in 3rd Normal Form.
3 A current state of the database:

### Factory

<table>
<thead>
<tr>
<th>FNr</th>
<th>FName</th>
<th>FStreet</th>
<th>FZip</th>
<th>FCity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Porsche Salzburg</td>
<td>Vogelweiderstrasse 12</td>
<td>A6020</td>
<td>Salzburg</td>
</tr>
<tr>
<td>2</td>
<td>Audi Graz</td>
<td>Kaerntnerstrasse 155</td>
<td>A8010</td>
<td>Graz</td>
</tr>
<tr>
<td>3</td>
<td>VW Linz</td>
<td>Industriezeile 209</td>
<td>A4010</td>
<td>Linz</td>
</tr>
</tbody>
</table>

### Employee

<table>
<thead>
<tr>
<th>ENr</th>
<th>FNr</th>
<th>EName</th>
<th>EPosition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>Aigner</td>
<td>manager</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Mueller</td>
<td>manager</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>Huber</td>
<td>clerk</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>Gruber</td>
<td>worker</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>Maier</td>
<td>manager</td>
</tr>
</tbody>
</table>

### Transfer

<table>
<thead>
<tr>
<th>TrNr</th>
<th>ENr</th>
<th>Date</th>
<th>SrcF</th>
<th>DestF</th>
<th>TNr</th>
<th>Qnt</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2002-01-07</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2002-01-08</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>2002-01-12</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2002-01-12</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

### Vehicular Inventory

<table>
<thead>
<tr>
<th>FNr</th>
<th>TNr</th>
<th>Qnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>70</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>150</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>TIdentifier</th>
<th>TData</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Volkswagen</td>
<td>family cars</td>
</tr>
<tr>
<td>2</td>
<td>Audi</td>
<td>sport cars</td>
</tr>
<tr>
<td>3</td>
<td>Skoda</td>
<td>sturdy cars</td>
</tr>
<tr>
<td>4</td>
<td>Seat</td>
<td>cheaper cars</td>
</tr>
<tr>
<td>5</td>
<td>Porsche</td>
<td>luxury cars</td>
</tr>
</tbody>
</table>
4 Database Queries (relational Algebra):

*Get the names of the factories, that have all types of vehicles in stock:*
project Vehicular_Inventory over FNr, TNr giving A;
project Type over TNr, giving B;
divide A by B giving C;
join C AND Factory over FNr giving Result;

*Get the names of the employees who made a transfer of Audi cars:*
join Employee AND Transfer over ENr giving A;
join A AND Type over TNr giving B;
select B where TIdentifier = “Audi” giving C;
project C over EName giving Result;

*Get the names of the employees who made a transfer with a quantity greater than 15 which is already complete:*
join Employee AND Transfer over ENr giving A;
join A AND Type over TNr giving B;
select B where Qnt > 15 giving C1;
select B where Complete = 1 giving C2;
C1 intersect C2 giving C;
project B over EName giving Result;
5 Database Queries (relational Calculus):

- Get the type-name(s) of the car(s) with the highest Quantity in stock of the Factory “Porsche Salzburg”:
  \[
  \exists F \exists V1 \forall V2 \left( V1.Qnt \geq V2.Qnt \land F.FName = \text{“Porsche Salzburg”} \land F.FNr = V1.FNr \land V1.TNr = T.TNr \right)
  \]

- Get the names of employees who made a transfer with less than 10 cars:
  \[
  \exists T \left( T.Qnt < 10 \land E.ENr = T.ENr \right)
  \]

- Get the names of employees who transferred any cars of the type “Skoda”:
  \[
  \exists E \exists T \exists Tr \left( T.TIdentifier = \text{“Skoda”} \land T.TNr = Tr.TNr \land Tr.ENr = E.ENr \right)
  \]
6 Database Queries (SQL):

- Get the names of the employees who transferred more than 15 cars:
  Select EName from Employee
  Where ENr in
    ( Select ENr From Transfer
      Group By ENr Having SUM (Qnt) > 15 )

- Get a list of type-names and type-data from cars that the employee “Aigner” transferred:
  Select TIdentifier, TData From Type
  Where TNr in
    ( Select TNr From Transfer
      Where ENr in
        ( Select ENr From Employee
          Where EName = “Aigner” ) )

- Get a list of all Factories that don’t have the car-type “Porsche” in stock:
  Select * From Factory
  Where FNr not in
    ( Select FNr From Vehicular_Inventory
      Where TNr in
        ( Select TNr From Type
          Where TIdentifier = “Porsche” ) )
7 Practical implementation of the database with mySQL

#Databases and Information Systems 1
#Constructional Example
#Database-Script
#Student: Alexander Rudolf Gruber
#Matricular Number: 9812938
SHOW DATABASES;
USE PROJECT

#Practical Definition of a database schema:
CREATE TABLE Factory (FNr Integer NOT NULL,
    FName Varchar(25) NOT NULL,
    FStreet Varchar(30),
    FZip Varchar(10),
    FCity Varchar(20),
    PRIMARY KEY(FNr));
DESCRIBE Factory;
CREATE TABLE Employee (ENr Integer NOT NULL,
    FNr Integer NOT NULL,
    EName Varchar(20) NOT NULL,
    EPosition Varchar(20) NOT NULL,
    PRIMARY KEY(ENr));
DESCRIBE Employee;
CREATE TABLE Type (TNr Integer NOT NULL,
    TIdentifier Varchar(20) NOT NULL,
    TData Varchar(50),
    PRIMARY KEY(TNr));
DESCRIBE Type;
CREATE TABLE Vehicular_Inventory (FNr Integer NOT NULL,
    TNr Integer NOT NULL,
    Qnt Integer NOT NULL,
    PRIMARY KEY (FNr, TNr));
DESCRIBE VEHICULAR_INVENTORY;
CREATE TABLE Transfer (TrNr Integer NOT NULL,
VO 506055: Constructional Example

ENr integer NOT NULL,
Date date NOT NULL,
SrcF integer NOT NULL,
DestF integer NOT NULL,
TNr integer NOT NULL,
Qnt integer NOT NULL,
Complete integer NOT NULL,
PRIMARY KEY (TrNr));

DESCRIBE TRANSFER;
SHOW TABLES;

# Creating a sample database content
LOAD DATA LOCAL INFILE "a:/factory.txt" INTO TABLE Factory;
LOAD DATA LOCAL INFILE "a:/Employee.txt" INTO TABLE Employee;
LOAD DATA LOCAL INFILE "a:/Type.txt" INTO TABLE Type;
LOAD DATA LOCAL INFILE "a:/Cars.txt" INTO TABLE Vehicular_Inventory;
LOAD DATA LOCAL INFILE "a:/Transfer.txt" INTO TABLE Transfer;

# Printing the sample database content out
SELECT * FROM Factory;
SELECT * FROM Employee;
SELECT * FROM TYPE;
SELECT * FROM VEHICULAR_INVENTORY;
SELECT * FROM TRAnSFER;

# Defining Database Queries

- # Select EName from Employee Where ENr in (Select ENr From Transfer Group By ENr Having SUM(Qnt) > 15);
- # Select TIdentifier, TData From Type Where TNr in (Select TNr From Transfer Where ENr in (Select ENr From Employee Where EName = “Aigner”));
- # Select * From Factory Where FNr not in ( Select FNr From Vehicular_Inventory Where TNr in ( Select TNr From Type Where TIdentifier = “Porsche” ) );

# Due to the fact, that subselects are not yet implemented for IN() functions, we have to rewrite the queries:

- Select Employee.EName From Employee, Transfer Where Transfer.ENr = Employee.ENr Group by Transfer.ENr Having SUM(Transfer.QNT) > 15;
• Select Employee.ENr From Employee, Transfer Where Transfer.ENr = Employee.ENr Group by Transfer.ENr Having SUM(Transfer.QNT) < 16;

• Select Type.TIdentifier, Type.TData From Type, Transfer, Employee Where Employee.ENr = "Aigner" And Employee.ENr = Transfer.ENr And Transfer.Tnr = Type.Tnr;

• Select Type.TIdentifier, Type.TData From Type, Transfer, Employee Where Employee.ENr = "Mueller" And Employee.ENr = Transfer.ENr And Transfer.Tnr = Type.Tnr;

• Select Factory.* From Factory, Vehicular_Inventory, Type Where Type.TIdentifier = "Porsche" And Type.TNr = Vehicular_Inventory.TNr And Vehicular_Inventory.FNr = Factory.Fnr;

• Select Factory.* From Factory, Vehicular_Inventory, Type Where Type.TIdentifier = "Skoda" And Type.TNr = Vehicular_Inventory.TNr And Vehicular_Inventory.FNr = Factory.Fnr;