2. Introductory SQL

2.13 SELECT all attributes

Get all data in the Student table:

SELECT * FROM Student;

2.14 SELECT all attributes

Get all data in the Course table:

SELECT * FROM Course;

2.15 SELECT only one attribute

Get the name of all students:

SELECT StudName FROM Student;

2.16 SELECT multiple attributes

Get the name and total credits of all students:

SELECT StudName, TotCredits FROM Student;

2.17 SELECT multiple attributes

Get the name, salary and department of all instructors:

SELECT InstName, Salary, DeptName FROM Instructor;

2.18 SELECT only some rows

Get the names of all students with a total credit of more than 100:

SELECT StudName FROM Student WHERE TotCredits > 100;

2.19 SELECT rows based on multiple conditions

Get the students in Computer Science with a total credit of more than 100:

SELECT StudName FROM Student WHERE DeptName = 'Comp. Sci.' AND TotCredits >100;

2.20 SELECT rows based on multiple conditions

Get the rooms with a capacity between 25 and 50, or located in the Painter building:

SELECT Room, Capacity FROM Classroom WHERE (Capacity >25 AND Capacity <50) OR (Building = 'Painter');

2.21 SELECT rows based on single condition

Get all department names not located in the Taylor building:

SELECT DeptName FROM Department WHERE Building <> 'Taylor';

2.22 SELECT based on two tables

What are the Course ID, year and grade for all courses taken by student Shankar:

SELECT CourseID, StudyYear, Grade FROM Takes, Student WHERE Takes.StudID=Student.StudID AND StudName = 'Shankar';

2.23 INSERT with multiple rows

Create two new Comp. Sci. courses CS-102 and CS-103 in table Course titled Weekly Seminar and Monthly Seminar, both with 0 credits:

INSERT Course VALUES ('CS-102', 'Weekly Seminar', 'Comp. Sci.', 0), ('CS-103', 'Monthly Seminar', 'Comp. Sci.', 0);

2.24 INSERT with multiple NULL values

Create a section for both CS-102 and CS-103 in Fall 2009, both with SectionID 1:

INSERT Section VALUES ('CS-102', 1, 'Fall', 2009, Null, Null, Null), ('CS-103', 1, 'Fall', 2009, Null, Null, Null);

2.25 INSERT with SELECT and NULL

In table Takes enroll every student in the Comp. Sci. department in the section for CS-102:

INSERT Takes SELECT StudID, 'CS-102', 1, 'Fall', 2009, Null FROM Student WHERE DeptName = 'Comp. Sci.';

2.26 DELETE

Delete both courses CS-102 and CS-103 in the Takes table:

DELETE FROM Takes WHERE CourseID = 'CS-102' OR CourseID = 'CS-103';

2.27 Update

Move the Finance department to the Taylor building.

```
UPDATE Department SET  Building = 'Taylor' WHERE
DeptName = 'Finance';
```

PS. Run the UniversityDB Script to restore tables to initial instances.

2.28 Create a Database

Write SQL DDL statements corresponding to the Relation Schemas below for an Insurance Database.

Person (DriverID, DriverName, Address)
Car (License, Model, ProdYear)
Accident (ReportNumber, AccDate, Location)
Owns (DriverID, License)
Participants (ReportNumber, License, DriverID, DamageAmount)

Make any reasonable assumptions about data types, and declare primary and foreign keys.

Solution.

```
CREATE DATABASE Insurance;
USE Insurance;

CREATE TABLE Person (
   DriverID CHAR(8),
   DriverName VARCHAR(45),
   Address VARCHAR(45),
   PRIMARY KEY (DriverID));

CREATE TABLE CAR (
   License CHAR(7),
   Model VARCHAR(45),
   ProdYear YEAR(4),
   PRIMARY KEY (License));

CREATE TABLE Accident (
   ReportNumber CHAR(10),
   AccDate DATE,
   Location VARCHAR(45),
   PRIMARY KEY (ReportNumber));

CREATE TABLE Owns (
   DriverID CHAR(8),
   License CHAR(7),
   PRIMARY KEY (DriverID , License),
   FOREIGN KEY (DriverID)
      REFERENCES Person (DriverID),
   FOREIGN KEY (License)
      REFERENCES Car (License));

CREATE TABLE Participants (
   ReportNumber CHAR(10),
   License CHAR(7),
   DriverID CHAR(8),
   DamageAmount DECIMAL(10 , 0 ),
   PRIMARY KEY (ReportNumber , License),
   FOREIGN KEY (License)
      REFERENCES Car (License),
   FOREIGN KEY (DriverID)
      REFERENCES Person (DriverID),
   FOREIGN KEY (ReportNumber)
      REFERENCES Accident (ReportNumber));
```

2.29 Populate a Database

Write SQL DML statements to populate the database with data, to end up with:

SELECT * FROM Person;

DriverID	DriverName	Address
31262549	Hans Hansen	Jernbane Alle 74, 2720 Vanløse

SELECT * FROM Car;

License	Model	ProdYear
JW46898	Honda Accord Aut. 2.0	2001

SELECT * FROM Accident;

ReportNumber	AccDate	Location
3004000121	2015-06-18	2605 Brøndby

SELECT * FROM Owns;

DriverID	License
31262549	JW46898

SELECT * FROM Participants;

ReportNumber	License	DriverID	DamageAmount
3004000121	JW46898	31262549	6800

Solution.

INSERT Person VALUES ('31262549', 'Hans Hansen','Jernbane Alle 74, 2720 Vanløse');

INSERT Car VALUES ('JW46898','Honda Accord Aut. 2.0',2001);

INSERT Accident VALUES ('3004000121', 20150618,'2605 Brøndby');

INSERT Owns VALUES ('31262549','JW46898');

INSERT Participants VALUES ('3004000121', 'JW46898','31262549',6800);