Chapter-8 Solution
Introduction to Structured Query Language (SQL)

1. Math the following clauses with their respective functions.

ALTER	Insert the values in the table
UPDATE	Restriction on columns
DELETE	Table definition
INSERT INTO	Change the name of column
CONSTRAINTS	Update existing information in a table
DESC	Delete an existing row from a table
CREATE	Create a database

Ans:

ALTER	Change the name of column
UPDATE	Update existing information in a table
DELETE	Delete an existing row from a table
INSERT INTO	Insert the values in the table
CONSTRAINTS	Restriction on columns
DESC	Table definition
CREATE	Create a database

2. Choose appropriate answer with respect to the following code snippet.

```
CREATE TABLE student (
    name CHAR(30),
    student_id INT,
    gender CHAR(1),
    PRIMARY KEY (student_id));
```

a. What will be the degree of student table?
   i. 30
   ii. 1
   iii. 3
   iv. 4

Ans: iii. 3 (degree means columns and total no of columns are 3)

b. What does 'name' represent in the above code snippet?

i. A table
ii. A row
iii. A column
iv. A database

Ans: iii. A column

c. What is true about the following SQL statement?

SelecT * fROM student;

i. Display content of table 'student'
ii. Display column names and content of table 'student'
iii. Results in error as improper case has been used
iv. Display only the column names of table 'student'

Ans: ii. Display column name and content of table 'student'.
(Because SQL is not case-sensitive)

d. What will be output of following query?

```
INSERT INTO student
VALUES ("Suhana", 109, 'F'),
VALUES ("Rivaan", 102, 'M'),
VALUES ("Atharv", 103, 'M'),
VALUES ("Rishika", 105, 'F'),
VALUES ("Garvit", 104, 'M'),
VALUES ("Shaurya", 109, 'M');
```

i) Error
ii) No Error
iii) Depends on Compiler
iv) Successful compilation of the query

Ans: i. Error
(Because we cannot use 'VALUE' keyword multiple times with single INSERT clause and also we are trying to insert value 109 two times in student_id column which is not possible because it is PRIMARY KEY)

e. In the following query how many rows will be deleted?

```
DELETE student
WHERE student_id =109;
```

i) 1 row
ii) All the rows where student ID is equal to 109

iii) No rows will be deleted
iv) 2 rows

Ans: i. 1 row
(Because student_id is a primary key and so it has all unique values.)

3. Fill in the blanks

a. _____ declares that an index in one table is related to that in another table.
   i. Primary key
   ii. Foreign Key
   iii. Composite Key
   iv. Secondary Key

Ans: ii. Foreign Key
(Because as it name implies 'Foreign' which means that field is related to field of another table. The key which is Primary key in another related table is called 'Foreign Key')

b. The symbol Asterisk (*) in a select query retrieves _____.
   i. All data from the table
   ii. Data of primary key only
   iii. NULL data
   iv. None of the mentioned

Ans: iv. None of the mentioned.
(Because * retrieves data of all attributes (columns) from a table not all data from table.)

4. Consider the following MOVIE database and answer the SQL queries based on it.

MovieID	MovieName	Category	ReleaseDate	ProductionCost	BusinessCost
001	Hindi_Movie	Musical	2018-04-23	124500	130000
002	Tamil_Movie	Action	2016-05-17	112000	118000
003	English_Movie	Horror	2017-08-06	245000	360000
004	Bengali_Movie	Adventure	2017-01-04	72000	100000
005	Telugu_Movie	Action	-	100000	-
006	Punjabo_Movie	Comedy	-	30500	-

a. Retrieve movies information without mentioning their column names.

Ans: SELECT * FROM MOVIE;

b. List business done by the movies showing only MovieID, MovieName and BusinessCost.

Ans: SELECT MovieID, MovieName, BusinessCost

FROM MOVIE

WHERE BusinessCost <> 0; [OR WHERE BusinessCost IS NOT NULL]

c. List the different categories of movies

Ans: SELECT DISTINCT Category FROM MOVIE;

d. Find the net profit of each movie showing its ID, Name and Net Profit. (***Hint: NetProfit = BusinessCost –ProductionCost)***
Make sure that the new column name is labeled as NetProfit. Is this column now a part of the MOVIE relation. If no, then what name coined for such columns? What can you say about the profit of a movie which has not yet released? Does your query result show profit as zero?

Ans: SELECT MovieID, MovieName, (BusinessCost – ProductionCost) AS NetProfit
FROM MOVIE
WHERE ReleaseDate IS NOT NULL;

(In this query we are finding the NetProfit by subtracting ProductionCost from BusinessCost and naming the subtraction as NetProft. Such columns which are not part of table are called **Generated Columns.** Because we cannot find the profit of movie which has not yet released, therefore we have set the criteria in WHERE clause to ReleaseDate IS NOT Null.

e. List all movies with ProductionCost greater than 80,000 and less than 1,25000 showing ID, Name and ProductionCost.

Ans: SELECT MovieID, MovieName, ProductionCost from
FROM MOVIE
WHERE ProductionCost BETWEEN 80000 AND 125000;

f. List all movies which fall in the category of Comedy or Action.

Ans: SELECT *
FROM MOVIE
WHERE Category IN ('Action', 'Comedy');

g. List the movies which have not been released yet.

Ans: SELECT *
FROM MOVIE
WHERE ReleaseDate IS NULL;

5. Suppose your school management has decided to conduct cricket matches between students of class XI and Class XII. Students of each class are asked to join any one of the four teams — Team Titan, Team Rockers, Team Magnet and Team Hurricane. During summer vacations, various matches will be conducted between these teams. Help your sports teacher to do the following:

a. Create a database "Sports".
Ans: CREATE DATABASE Sports;

b. Create a table "TEAM" with following considerations:
   i. It should have a column TeamID for storing an integer value between 1 to 9, which refers to unique identification of a team.
   ii. Each TeamID should have its associated name (TeamName), which should be a string of length not less than 10 characters.

Ans: CREATE TABLE TEAM (
TeamID INT CHECK (TeamID BETWEEN 1 AND 9),
TeamName VARCHAR(20)
CHECK (LENGTH (TeamName) > 10));

c. Using table level constraint, make TeamID as primary key.
Ans: ALTER TABLE TEAM
ADD PRIMARY KEY (TeamID);

d. Show the structure of the table TEAM using SQL command.
Ans: DESC TEAM; or DESCRIBE TEAM;

e. As per the preferences of the students four teams were formed as given below. Insert these four rows in TEAM table:
Row 1: (1, Team Titan)
Row 2: (2, Team Rockers)
Row 3: (3, Team Magnet)
Row 4: (4, Team Hurricane)
Ans: INSERT INTO TEAM
VALUES
(1, "Team Titan") ,
(2, "Team Rockers") ,
(3, "Team Magnet") ,
(4, "Team Hurricane");

f. Show the contents of the table TEAM.

Ans: SELECT * FROM TEAM;

g. Now create another table below. MATCH_DETAILS and insert data as shown in table. Choose appropriate domains and constraints for each attribute.

MatchID	MatchDate	FirstTeamID	SecondTeamID	FirstTeamScore	SecondTeamScore
M1	2018-07-17	1	2	90	86
M2	2018-07-18	3	4	45	48
M3	2018-07-19	1	3	78	56
M4	2018-07-19	2	4	56	67
M5	2018-07-20	1	4	32	87
M6	2018-07-21	2	3	67	51

Ans:
```
CREATE TABLE MATCH_DETAILS (
MatchID CHAR(2) PRIMARY KEY,
MatchDate DATE,
FirstTeamID INT,
SecondTeamID INT,
FirstTeamScore INT,
secondTeamScore INT);
```

h. Use the foreign key constraint in the MATCH_ DETAILS table with reference to TEAM table so that MATCH_DETAILS table records score of teams existing in the TEAM table only.

Ans:
```
ALTER TABLE MATCH_DETAILS
ADD FOREIGN KEY (FirstTeamID) REFERENCES TEAM(TeamID),
ADD FOREIGN KEY (SecondTeamID) REFERENCES TEAM(TeamID);
```

6. Using the sports database containing two relations (TEAM, MATCH_DETAILS), answer the following relational algebra queries.

a. Retrieve the MatchID of all those matches where both the teams have scored > 70.

Ans:
```
SELECT MatchID FROM MATCH_DETAILS
WHERE FirstTeamScore >70 AND SecondTeamScore >70;
```

b. Retrieve the MatchID of all those matches where FirstTeam has scored < 70 but SecondTeam has scored > 70.

Ans:
```
SELECT MatchID FROM MATCH_DETAILS
WHERE FirstTeamScore <70 AND SecondTeamScore >70;
```

c. Find out the MatchID and date of matches played by Team 1 and won by it.
Ans: SELECT MatchID, MatchDate FROM MATCH_DETAILS
WHERE FirstTeamID = 1 AND FirstTeamScore > SecondTeamScore;

d. Find out the MatchID of matches played by Team 2 and not won by it.
Ans: SELECT MatchID FROM MATCH_DETAILS
WHERE FirstTeamID = 2 AND FirstTeamScore > SecondTeamScore;

e. In the TEAM relation, change the name of the relation to T_DATA. Also change the attributes TeamID and TeamName to T_ID and T_NAME respectively.

Ans: RENAME TABLE TEAM TO T_DATA;

ALTER TABLE TEAMS
RENAME TeamID T_ID INT,
RENAME TeamData T_DATA VARCHAR(20);

7. Differentiate between the following commands:

a. ALTER and UPDATE
Ans:

ALTER	UPDATE
ALTER is a DDL statement	UPDATE is a DML statement
ALTER is a SQL command that is used to modify, delete or add a column to an existing table in a database.	UPDATE is a SQL command that is used to update existing records in a database
ALTER command modifies the database schema (structure)	UPDATE statement only modifies records in a database without modifying its structure.

b. DELETE and DROP
Ans:

DELETE	DROP
DELETE is DML statement	DROP is DDL statement
DELETE is a SQL Command, removes some or all records from a table	DROP is a SQL Command, which removes different elements of database like table, constraints or entire database schema.
WHERE clause can be used with DELETE	WHERE can not be used with DROP

8. Create a database called STUDENT_PROJECT having the following tables. Choose appropriate data type and apply necessary constraints.
Table: STUDENT

RollNo	Name	Stream	Section	RegistrationID

*The values in Stream column can be either Science, Commerce, or Humanities.
* The values in Section column can be either I or II.

Ans: CREATE DATABASE STUDENT_PROJECT;

```
CREATE TABLE STUDENT (
RollNo INT NOT NULL UNIQUE,
Name varchar(20) NOT NULL,
Stream varchar(20) CHECK (Stream IN ('Science', 'Commerce', 'Humanities')),
Section varchar(2) CHECK (Section IN ('I', 'II')),
RegistrationID int PRIMARY KEY);
```

Table: PROJECT_ASSIGNED

RegistrationID	ProjectID	AssignDate

Ans:
```
CREATE TABLE PROJECT_ASSIGNED (
RegistrationID INT,
ProjectID INT,
AssignDate DATE,
CONSTRAINT FK_STUDENT FOREIGN KEY (RegistrationID)
REFERENCES STUDENT (RegistrationID),
CONSTRAINT FK_PROJECT FOREIGN KEY (ProjectID) REFERENCES
PROJECT (ProjectID));
```

Table: PROJECT

ProjectID	PreojectName	SubmissionDate	TeamSize	GuideTeacher

Ans:
```
CREATE TABLE PROJECT (
ProjectID INT PRIMARY KEY,
ProjectName VARCHAR (20) NOT NULL,
SubmissionDate DATE,
TeamSize INT,
GuideTeacher VARCHAR (30));
```

a. Populate these tables with appropriate data.
Ans: Use INSERT INTO statement.

b. Write SQL queries for the following.
c. Find the names of students in Science Stream.

Ans: SELECT Name from STUDENT
WHERE Stream = "Science";
(We can alse write WHERE Stream In('Scince'))

d. What will be the primary keys of the three tables?
Ans: Following are the Primary keys of the three tables

Table	PrimaryLey
STUDENT	RegistrationID
PROJECT	ProjectID
PROJECT_ASSIGNED	-

e. What are the foreign keys of the three relations?
Ans: RegistrationID of STUDENT relation and ProjectID of PROJECT relation are Foreign Key in PROJECT_ASSIGNED relation.

f. Finds names of all the students studying in class 'Commerce stream' and are guided by same teacher, even if they are assigned different projects.
Ans:
SELECT Name FROM STUDENT, PROJECT, PROJECT_ASSGINED
WHERE (STUDENT.RegistrationID = PROJECT_ASSIGNED.RegistrationID
AND PROJECT.ProjectID = PROJECT_ASSIGNED.ProjectID) AND
(Stream = "Commerce Stream" AND GuideTeacher = "teachername");

9. An organization ABC maintains a database EMP-DEPENDENT to record the following details about its employees and their dependents.

EMPLOYEE (AadhaarNo, Name, Address, Department, EmpID)
DEPENDENT (EmpID, DependentName, Relationship)
Use the EMP-DEPENDENT database to answer the following SQL queries:

a. Find the names of employees with their dependent names.
Ans: Select E.Name, D.DependentName
FROM EMPLOYEE E, DEPENDENT D
WHERE E.EmpID = D.EmpID;

b. Find employee details working in a department, say, 'PRODUCTION'.

Ans: SELECT * FROM EMPLOYEE WHERE Department = 'PRODUCTION';

c. Find employee names having no dependent

Ans: SELECT Name FROM EMPLOYEE
WHERE EmpID NOT IN (SELECT EMPID FROM DEPENDENT);

d. Find names of employees working in a department, say, 'SALES' and having exactly two dependents.

Ans: SELECT E.Name FROM EMPLO.YEE E, DEPENDENT D
WHERE E.EmpID = D.EmpID AND Department = 'SALES'
GROUP BY E.EmpID
HAVING COUNT (D.EmpID) = 2;

10. A shop called Wonderful Garments that sells school uniforms maintain a database SCHOOL_UNIFORM as shown below. It consisted of two relations — UNIFORM and PRICE. They made UniformCode as the primary key for UNIFORM relation. Further, they used UniformCode and Size as composite keys for PRICE relation. By analyzing the database schema and database state, specify SQL queries to rectify the following anomalies.

UNIFORM		
UCode	UName	Ucolor
1	Shirt	White
2	Pant	Grey
3	Skirt	Grey
4	Tie	Blue
5	Socks	Blue
6	Belt	blue

PRICE		
UCode	Size	Price
1	M	500
1	L	580
1	XL	620
2	M	810
2	L	940
2	XL	940
3	M	770
3	L	830
3	XL	910
4	S	150
4	L	170
5	S	180
5	L	210
6	M	110
6	L	140
6	XL	160

a. The PRICE relation has an attribute named Price. In order to avoid confusion, write SQL query to change the name of the relation PRICE to COST.

Ans: ALTER TABLE PRICE RENAME TO COST;

b. M/S Wonderful Garments also keeps handkerchiefs of red color, medium size of 100 each. Insert this record in COST table.
Ans: INSERT INTO UNIFORM VALUES (7, 'Handkerchief', 'Red')
INSERT INTO COST VALUES (7, 'M', 100);

c. When you used the above query to insert data, you were able to enter the values for handkerchief without entering its details in the UNIFORM relation. Make a provision so that the data can be entered in COST table only if it is already there in UNIFROM table.

Ans: ALTER TABLE COST ADD FOREIGN KEY (UCode)
REFERENCES UNIFORM (UCode);

d. Further, you should be able to assign a new UCode to an item only if it has a valid Uname. Write a query to add appropriate constraint to the SCHOOL_ UNIFORM database.
Ans: ALTER TABLE UNIFORM ADD UNIQUE (UName);

e. ALTER table to add the constraint that price of an item is always greater than zero.
Ans: ALTER TABLE COST ADD CHECK (Price >100);