CHAPTER 5

Subquery

ASSESSMENTS/ACTIVITIES:

Use the table below and create a program for the following problems. Please create the table and populate it for your test data for these exercises.

EMPLOYEE (EmpNo, Sname, JobLevel, Status, DateHired, Salary, EmpAddress)

JobLevel values: MGR - manager, RF - rank and file, S – supervisor

Status values: R - regular, P - probationary, C – contractual

```
CREATE DATABASE activity5;

CREATE TABLE Employee (
    EmpNo INTEGER NOT NULL,
    Sname VARCHAR(30) NOT NULL,
    JobLevel VARCHAR(30),
    JobStatus CHAR,
    DateHired DATE,
    Salary FLOAT,
    EmpAddress VARCHAR(30),
    CONSTRAINT emp_pk PRIMARY KEY (EmpNo)
);
```

```
INSERT INTO Employee VALUES
(11111, 'Cara Dulay', 'MGR', 'R', '2010-11-21', '10000', 'Manila'),
(11112, 'Ian Bartolome', 'RF', 'R', '2020-06-12', '2500', 'Caloocan'),
(11123, 'Agatha Tagala', 'S', 'R', '2018-01-17', '100000', 'Quezon'),
(11234, 'Andrew Castillo', 'MGR', 'R', '2011-06-21', '10000', 'Pasig'),
(12345, 'Luis Villanueva', 'RF', 'R', '2011-11-21', '3000', 'Pasig'),
(00001, 'Patrick Sudaria', 'MGR', 'P', '2022-04-10', '1000', 'Pasig'),
(00002, 'John Ignacio', 'RF', 'P', '2020-12-12', '1500', 'Manila'),
(00003, 'Irvin Presto', 'S', 'P', '2019-05-12', '50000', 'Caloocan'),
(00004, 'Jeri Claveria', 'MGR', 'P', '2021-03-10', '1000', 'Manila'),
(00005, 'Michaella Soriano', 'RF', 'P', '2022-09-12', '1500', 'Paranaque'),
(10000, 'Elora Nava', 'MGR', 'C', '2018-01-12', '10000', 'Paranaque'),
(20000, 'Diosa Gutierrez', 'RF', 'C', '2015-02-13', '3000', 'Mandaluyong'),
(30000, 'Elaine Cabezudo', 'S', 'C', '2015-02-13', '50000', 'Manila'),
(40000, 'Julia Boco', 'RF', 'C', '2015-02-27', '3000', 'Mandaluyong'),
(50000, 'Elaine Suganob', 'S', 'C', '2015-08-13', '50000', 'Quezon');
```

1. List the records of employees whose salary is the same as employee whose empno is '11111'.

```
SELECT *
FROM Employee
WHERE Salary = (SELECT Salary FROM Employee WHERE EmpNo = 11111);
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
10000	Elora Nava	MGR	C	2018-01-12	10000	Paranaque
11111	Cara Dulay	MGR	R	2010-11-21	10000	Manila
11234	Andrew Castillo	MGR	R	2011-06-21	10000	Pasig
NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. List the names of employees whose salary is greater than the minimum salary.

```
SELECT *
FROM Employee
WHERE Salary > (SELECT MIN(SALARY) FROM Employee);
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
2	John Ignacio	RF	P	2020-12-12	1500	Manila
3	Irvin Presto	S	P	2019-05-12	50000	Caloocan
5	Michaella Soriano	RF	P	2022-09-12	1500	Paranaque
10000	Elora Nava	MGR	C	2018-01-12	10000	Paranaque
11111	Cara Dulay	MGR	R	2010-11-21	10000	Manila
11112	Ian Bartolome	RF	R	2020-06-12	2500	Caloocan
11123	Agatha Tagala	S	R	2018-01-17	100000	Quezon
11234	Andrew Castillo	MGR	R	2011-06-21	10000	Pasig
12345	Luis Villanueva	RF	R	2011-11-21	3000	Pasig
20000	Diosa Gutierrez	RF	C	2015-02-13	3000	Mandaluyong
30000	Elaine Cabezudo	S	C	2015-02-13	50000	Manila
40000	Julia Boco	RF	C	2015-02-27	3000	Mandaluyong
50000	Elaine Suganob	S	C	2015-08-13	50000	Quezon
NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. List the names of the employees who have the highest salaries.

```
SELECT *
FROM Employee
WHERE Salary = (SELECT MAX(SALARY) FROM Employee);
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
11123	Agatha Tagala	S	R	2018-01-17	100000	Quezon
NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. Create another table called Regular_Emp, with the same attributes except for status.

   REGULAR_EMP (EmpNo, Sname, JobLevel, DateHired, Salary, EmpAddress)

   4.1 Copy all records from Employee table to Regular_Emp table if the status is ='R' This will effectively store in Regular_Emp table all records of regular employees.

```
CREATE TABLE Regular_Emp (
    EmpNo INTEGER NOT NULL,
    Sname VARCHAR(30) NOT NULL,
    JobLevel VARCHAR(30),
    DateHired DATE,
    Salary FLOAT,
    EmpAddress VARCHAR(30),
    CONSTRAINT emp_pk PRIMARY KEY (EmpNo)
);

INSERT INTO Regular_Emp (EmpNo, Sname, JobLevel, DateHired, Salary, EmpAddress)
SELECT EmpNo, Sname, JobLevel, DateHired, Salary, EmpAddress
FROM Employee
WHERE JobStatus = 'R';
```

5. List all records of employees whose salary is less than the average salary of all employees.

```
SELECT *
FROM Employee
WHERE Salary < (SELECT AVG(SALARY) FROM Employee);
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
1	Patrick Sudaria	MGR	P	2022-04-10	1000	Pasig
2	John Ignacio	RF	P	2020-12-12	1500	Manila
4	Jeri Claveria	MGR	P	2021-03-10	1000	Manila
5	Michaella Soriano	RF	P	2022-09-12	1500	Paranaque
10000	Elora Nava	MGR	C	2018-01-12	10000	Paranaque
11111	Cara Dulay	MGR	R	2010-11-21	10000	Manila
11112	Ian Bartolome	RF	R	2020-06-12	2500	Caloocan
11234	Andrew Castillo	MGR	R	2011-06-21	10000	Pasig
12345	Luis Villanueva	RF	R	2011-11-21	3000	Pasig
20000	Diosa Gutierrez	RF	C	2015-02-13	3000	Mandaluyong
40000	Julia Boco	RF	C	2015-02-27	3000	Mandaluyong
NULL	NULL	NULL	NULL	NULL	NULL	NULL

CHAPTER 6

Transaction

ASSESSMENTS/ACTIVITIES:

Answer the following questions:

1. Explain what a transaction is.

   - **Transactions combine a group of tasks into a single execution unit.** Each transaction starts with a specific task and ends when all of the tasks in the group are completed successfully. The transaction fails if any of the tasks fails.

2. Explain atomicity and durability.

   - **Atomicity** property ensures that all DML Operations (i.e., insert, update, delete) within a single transaction are either executed successfully or not executed at all.
   - **Durability** property ensures that the database keeps track of pending changes so that the system can recover in the event of further operating system and application failures.

3. What is the default mode of transaction?
   - **Auto-commit Transaction Mode** is the default transaction mode. Each SQL statement is evaluated as a transaction, and successful statements are immediately committed, while failed statements are immediately rolled back.

4. How do you begin an explicit transaction?

   - To begin an explicit transaction, the programmer must explicitly write a **START and END statement.**

5. Discuss serializability in the context of isolation as a property of a transaction.

   - A transaction runs in complete isolation in a serializable Isolation level. A serializable transaction operates in an environment that **appears as if no other users are modifying data in the application (database), but there are two concurrent transactions occurring** within the database at the same time that are isolated from one another.

6. Use the same table and data used in previous activities EMPLOYEE(EmpNo, Sname, JobLevel, Status, DateHired, Salary, EmpAddress).

   6.1 Use explicit transaction mode.

6.2 Update any record in the file.

```
USE activity5;

START TRANSACTION;
UPDATE EMPLOYEE
SET
    Salary = 1000
WHERE
    EmpNo = 1 AND SALARY = 2000;

COMMIT;
ROLLBACK;
```

Before Commit:

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
1	Patrick Sudaria	MGR	P	2022-04-10	1000	Pasig

After Commit

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
1	Patrick Sudaria	MGR	P	2022-04-10	2000	Pasig

After Rollback

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
1	Patrick Sudaria	MGR	P	2022-04-10	1000	Pasig

6.1 If you do not COMMIT nor ROLLBACK, what happens?

- The transaction will continue to exist indefinitely.

CHAPTER 7

Stored Procedure

ASSESSMENTS/ACTIVITIES:

Answer the following questions:

1. Give one advantage of a stored procedure and explain.
   - **Reusable:** By simply calling it, multiple users and applications can easily use and reuse stored procedures.
2. Where do you think will a stored procedure be most useful?
   - When processing large amounts of data on the server, as well as for database triggers that cannot be done in code. Also, when you need to perform operations on data that does not need to leave the database
3. Why does the use of a stored procedure decrease network traffic?
   - When a stored procedure is called, only the procedure call is sent to the server, not the statements contained within the procedure.
4. Use the same table and data used in previous activities.

   EMPLOYEE (EmpNo, Sname, JobLevel, Status, DateHired, Salary, EmpAddress)

   4.1 Create a stored procedure that will display all records from Employee table given an input parameter Status. So, if the input value is 'P', all probationary employees will be displayed, if the input value is 'R', all regular employees' records will be displayed, and so on.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `filter_employee_status`(
    IN x CHAR(1)
)
BEGIN
    SELECT *
    FROM EMPLOYEE
    WHERE JobStatus = x;
END
```

```
CALL filter_employee_status('P');
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
1	Patrick Sudaria	MGR	P	2022-04-10	2000	Pasig
2	John Ignacio	RF	P	2020-12-12	1500	Manila
3	Irvin Presto	S	P	2019-05-12	50000	Caloocan
4	Jeri Claveria	MGR	P	2021-03-10	1000	Manila
5	Michaella Soriano	RF	P	2022-09-12	1500	Paranaque

```
CALL filter_employee_status('R');
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
11111	Cara Dulay	MGR	R	2010-11-21	10000	Manila
11112	Ian Bartolome	RF	R	2020-06-12	2500	Caloocan
11123	Agatha Tagala	S	R	2018-01-17	100000	Quezon
11234	Andrew Castillo	MGR	R	2011-06-21	10000	Pasig
12345	Luis Villanueva	RF	R	2011-11-21	3000	Pasig

```
CALL filter_employee_status('C');
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
10000	Elora Nava	MGR	C	2018-01-12	10000	Paranaque
20000	Diosa Gutierrez	RF	C	2015-02-13	3000	Mandaluyong
30000	Elaine Cabezudo	S	C	2015-02-13	50000	Manila
40000	Julia Boco	RF	C	2015-02-27	3000	Mandaluyong
50000	Elaine Suganob	S	C	2015-08-13	50000	Quezon

4.2 Create a stored procedure that will list all records for an employee with a given hire date.

```
CALL filter_employee_hireddate('2015-02-27');
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
40000	Julia Boco	RF	C	2015-02-27	3000	Mandaluyong

4.3 Create a stored procedure that will list the names of those who are hired for the current day (no input parameter here, check out a scalar function that can be used).

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `filter_employee_hireddate_range`()
BEGIN
SELECT *
FROM EMPLOYEE
WHERE DateHired <= (CURRENT_DATE()+0);
END
```

```
CALL filter_employee_hireddate_range();
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
1	Patrick Sudaria	MGR	P	2022-04-10	2000	Pasig
2	John Ignacio	RF	P	2020-12-12	1500	Manila
3	Irvin Presto	S	P	2019-05-12	50000	Caloocan
4	Jeri Claveria	MGR	P	2021-03-10	1000	Manila
5	Michaella Soriano	RF	P	2022-09-12	1500	Paranaque
10000	Elora Nava	MGR	C	2018-01-12	10000	Paranaque
11111	Cara Dulay	MGR	R	2010-11-21	10000	Manila
11112	Ian Bartolome	RF	R	2020-06-12	2500	Caloocan
11123	Agatha Tagala	S	R	2018-01-17	100000	Quezon
11234	Andrew Castillo	MGR	R	2011-06-21	10000	Pasig
12345	Luis Villanueva	RF	R	2011-11-21	3000	Pasig
20000	Diosa Gutierrez	RF	C	2015-02-13	3000	Mandaluyong
30000	Elaine Cabezudo	S	C	2015-02-13	50000	Manila
40000	Julia Boco	RF	C	2015-02-27	3000	Mandaluyong
50000	Elaine Suganob	S	C	2015-08-13	50000	Quezon

CHAPTER 8

Views

ASSESSMENTS/ACTIVITIES:

Answer the following questions:

1. State an example where a view simplifies a query.
   - When you have a long and complex query, you can save it in a view and perform the SELECT statement on that view to save time retyping and running the query.
2. Give an example where you can apply the use of a view for security purposes
   - We have a table with confidential and sensitive company information. The admins should not have direct access to the table for editing or data security purposes and should only be able to see the details that they are permitted to see.
3. Use the same table and data used in previous activities

   EMPLOYEE (EmpNo, Sname, JobLevel, Status, DateHired, Salary, EmpAddress)

   Provide your own name for the views to be created.

   3.1 Create a view that will display only empno, datehired, salary of records from the Employee table if the employee salary is equal to the highest salary.

```
CREATE VIEW HighestPerformingEmployee  AS
SELECT EmpNo, DateHired, Salary
FROM EMPLOYEE
WHERE Salary = (SELECT MAX(SALARY) FROM Employee);

SELECT *
FROM HighestPerformingEmployee;
```

EmpNo	DateHired	Salary
11123	2018-01-17	100000

   3.2 Create a view that will display only empno, datehired, salary of records from the Employee table if the employee salary is in the highest salary range. Highest salary range is: highest salary - 5000 to highest salary.

   For example, if the highest salary is 20,000. The highest salary range is 15,000 to 20,000. The lower range was taken by subtracting 5000 from the highest salary.

   There's no between 95,000-100,000

```sql
CREATE VIEW HighestPerformingEmployeeRange  AS
SELECT EmpNo, DateHired, Salary
FROM EMPLOYEE
WHERE  Salary > ((SELECT MAX(SALARY) FROM Employee) - 5000) OR Salary < (SELECT MAX(SALARY) FROM Employee);

SELECT *
FROM HighestPerformingEmployeeRange;
```

CHAPTER 9

Triggers

ASSESSMENTS/ACTIVITIES:

Answer the following questions:

1. When do you use a BEGIN…END in the trigger body?
   - When grouping the multiple statement that to be used and executed
2. What is the difference between a DML and a DDL trigger?
   - DML triggers are triggered when data is modified by the user.
   - DDL triggers are triggered when a database structure is created, changed or dropped from the database.
3. What alias do you use to get column values of deleted records?
   - FROM **DELETED**
4. Use the same table and data used in previous activities.

   EMPLOYEE (EmpNo, Sname, JobLevel, Status, DateHired, Salary, EmpAddress)

   4.1 Create another table and call it EMPLOYEE_HIST. This will be the history table or audit trail for all changes to Employee table. It should have the same fields as Employee.

```
CREATE TABLE EMPLOYEE_HIST(
    EmpNo INTEGER NOT NULL,
    Sname VARCHAR(30) NOT NULL,
    JobLevel VARCHAR(30),
    JobStatus CHAR,
    DateHired DATE,
    Salary FLOAT,
    EmpAddress VARCHAR(30),
    CONSTRAINT emp_pk PRIMARY KEY (EmpNo)
);
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
NULL	NULL	NULL	NULL	NULL	NULL	NULL

   4.2 Create a trigger that will store in Employe_Hist table all records which are deleted from Employee table.

```
CREATE
    TRIGGER  EMPLOYEE_DELETED
 AFTER DELETE ON EMPLOYEE FOR EACH ROW
    INSERT INTO EMPLOYEE_HIST VALUES (OLD.EmpNo , OLD.Sname , OLD.JobLevel ,
                                      OLD.JobStatus , OLD.DateHired , OLD.Salary, OLD.EmpAddress);

                                      DELETE FROM EMPLOYEE
                                      WHERE EmpNo = 11111;
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
11111	Cara Dulay	MGR	R	2010-11-21	10000	Manila
NULL	NULL	NULL	NULL	NULL	NULL	NULL

EMPLOYEE HIST

10000	Elora Nava	MGR	C	2018-01-12	10000	Paranaque
11112	Ian Bartolome	RF	R	2020-06-12	2500	Caloocan
11123	Agatha Tagala	S	R	2018-01-17	100000	Quezon
11234	Andrew Castillo	MGR	R	2011-06-21	10000	Pasig
12345	Luis Villanueva	RF	R	2011-11-21	3000	Pasig

EMPLOYEE after deleting 1111

10000	Elora Nava	MGR	C	2018-01-12	10000	Paranaque
11111	Cara Dulay	MGR	R	2010-11-21	10000	Manila
11112	Ian Bartolome	RF	R	2020-06-12	2500	Caloocan
11123	Agatha Tagala	S	R	2018-01-17	100000	Quezon
11234	Andrew Castillo	MGR	R	2011-06-21	10000	Pasig
12345	Luis Villanueva	RF	R	2011-11-21	3000	Pasig

EMPLOYEE before deleting 1111

4.1 Create a trigger that will store in Employe_Hist table the before and after images of all records which are updated in Employee table

```
CREATE
    TRIGGER  EMPLOYEE_BEFDELETED
 BEFORE UPDATE ON EMPLOYEE FOR EACH ROW
    INSERT INTO EMPLOYEE_HIST VALUES (OLD.EmpNo , OLD.Sname , OLD.JobLevel ,
                                      OLD.JobStatus , OLD.DateHired , OLD.Salary , OLD.EmpAddress);

CREATE
    TRIGGER  EMPLOYEE_AFDELETED
 AFTER UPDATE ON EMPLOYEE FOR EACH ROW
    INSERT INTO EMPLOYEE_HIST VALUES (NEW.EmpNo , NEW.Sname , NEW.JobLevel ,
                                      NEW.JobStatus , NEW.DateHired , NEW.Salary , NEW.EmpAddress);
```

```
UPDATE EMPLOYEE
SET JobStatus = 'R'
WHERE EmpNo = 40000;
```

EmpNo	Sname	JobLevel	JobStatus	DateHired	Salary	EmpAddress
11111	Cara Dulay	MGR	R	2010-11-21	10000	Manila
40000	Julia Boco	RF	C	2015-02-27	3000	Mandaluyong
NULL	NULL	NULL	NULL	NULL	NULL	NULL

UPDATING EMP NO. 40000 TO JOBSTATUS ‘R’ AND CAN BE SEEN THE OLD RECORD IN EMPLOYEE_HIST

5. What events will fire a DDL trigger (for SQL Server)?
    - CREATE_TABLE
    - ALTER_TABLE
    - DROP_TABLE
    - CREATE_VIEW
    - ALTER_VIEW
    - DROP_VIEW

CHAPTER 10

Error Handling

ASSESSMENTS/ACTIVITIES:

Answer the following questions:

1. Why are error handling routines used in programming?

    - Error handling is important because it informs end users of your code that something has gone wrong and that they should contact technical support, or that someone from tech support has been notified and knows what the next step should be. Error handling makes your code easier to maintain and you can get a report of exactly how the bug occurred so you can fix it.