PROJECT MANUAL

"Database Management system"

CC-215

Database for a Dental Clinic

Submitted to :

Prof. Sehrish khan

Submitted by:

Neelam (110852)

Shanzay(110828)

TABLE OF CONTENTS

DESCRIPTION OF PROJECT DATABASE

Treatments like root canals span several appointments. Each visit logs performed procedures, tools used, medications prescribed, and pain level reported. Equipment usage is tracked to ensure sterilization logs are complete. Follow-up visits are auto-scheduled, and patients receive reminders. Equipment nearing end-of-life is flagged for replacement.

Entities along with their attributes:

- **Treatment:**
  1. Treatment_id (PK)
  2. Name
  3. Description
  4. Patient_id (FK)
  5.Dentist_id(FK)

- **Patient:**
  1. Patient_id (PK)
  2. DOB
  3. Phone_no (multivalue attribute)
  4. Name (composite attribute)
  5. Reminder_id (FK)

6. Visit_id (FK)
7. Dentist_id (FK)
8. Treatment_id (FK)

- **Dentist:**
  1. Dentist_id (PK)
  2. Name (composite)
  3. Specialization (multivalue attribute)
  4. Patient_id (FK)
  5. Visit_id(FK)

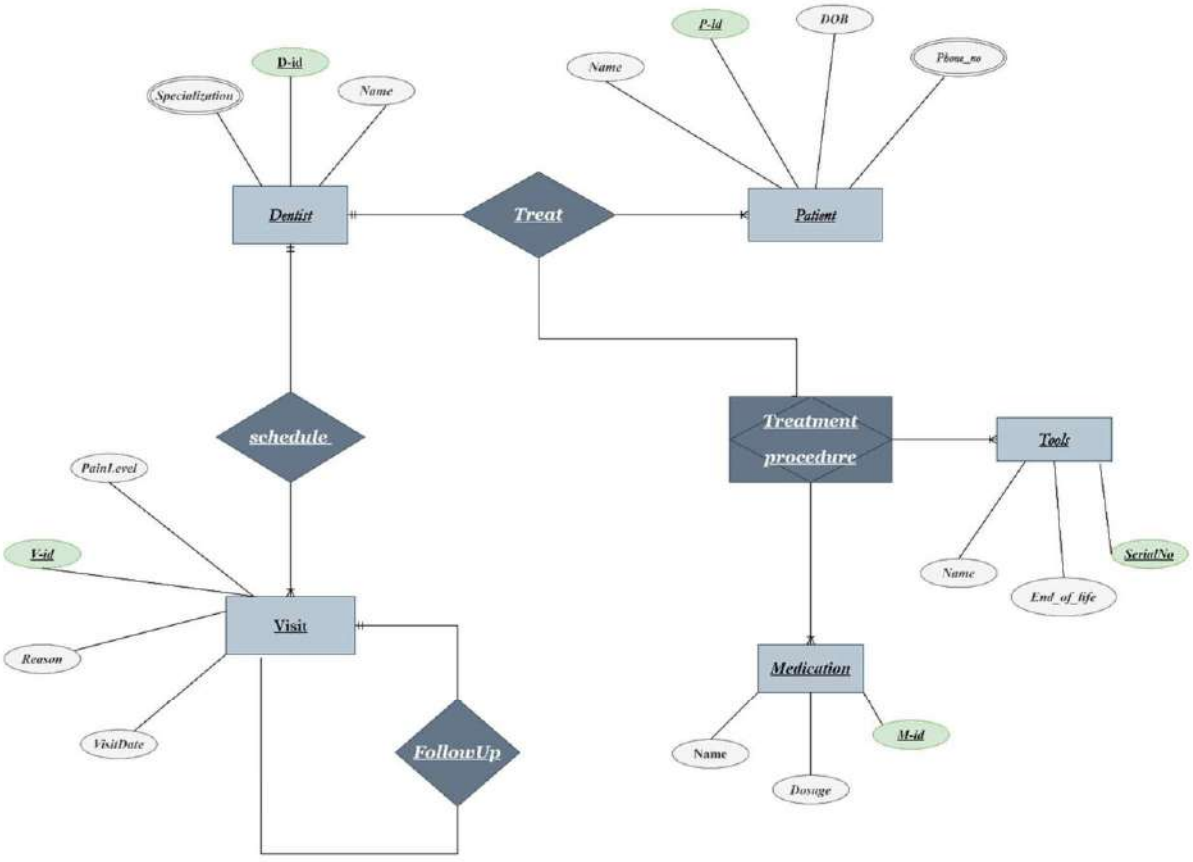
- **Reminder:**
  1. Reminder_id (PK)
  2. Date
  3. Message
  4. Visit_id(FK)
  5. Patient_id(FK)

- **Tool:**
  1. Serial_no(PK)
  2. End_of_life
  3. Name
  4. Sterilisation_id(FK)
  5. Visit_id(FK)

- **Medication:**
  1. M_ID(PK)
  2. Name
  3. Dosage
  4. Visit_id(FK)
- **Procedure:**
  1. P_id (PK)
  2. Name
  3. Description
  4. Visit_id(FK)

- **Visit:**
  1. Visit_id(PK)
  2. P_id (FK)
  3. Tool_id (FK)
  4. M_id (FK)
  5. Reminder_id(FK)
  6. Patient_id(FK)
  7. Dentist_id(FK)
  8. VisitDate
  9. Pain_level
  10. Reason

ERD OF DENTAL CLINIC DATABASE



Dental_Clinic DB

SCHEMA DESIGN

1. Dentist (d_id, Name)
   Specialisation (d_id , name)
   Patient (p_id, name, DOB,d_id)
   Phone (phone_no, p_id)

2. Dentist (d_id, Name)
   Specialization (d_id ,name)
   Visit (v_id, Pain_level, Reason, VisitDate, d_id)

3. Dentist (d_id, Name)
   Specialization (d_id ,name)
   Treatment_procedure (d_id, serial_no)
   Tools (serial_no , End_of_Life , Name)

4. Dentist (d_id, Name)
   Specialization (d_id , name)
   Treatment_procedure (d_id, M_id)
   Medication (M_id ,Dosage, name)

5. Patient (p_id, name, DOB)
   Phone (phone_no, p_id)
   Treatment_procedure (p_id, M_id)
   Medication (M_id ,Dosage, name)

6. Patient (p_id, name, DOB)
   Phone (phone_no, p_id)

Treatment_procedure (p_id, serial_no)

Tools (serial_no , End_of_Life , Name)

NORMALIZATION OF DENTAL CLINIC DATABASE

- The database schema was carefully reviewed and found to already following the principles of normalization up to the Third Normal Form (3NF).

- Each table contains atomic values, ensuring that all attributes store indivisible data.

- No partial dependencies exist, as all non-key attributes are fully functionally dependent on the primary key.

- Moreover, there are no transitive dependencies between non-key attributes.

- As a result, the database design eliminates data redundancy and ensures data integrity.

- The relationships between tables, such as one-to-many and many-to-one associations, have been established to maintain consistency and prevent anomalies during data operations.

The following entities and tables demonstrate normalization principles in a dental clinic database:

1. **Patient:** p_id, patient_name
2. **Patient Phone:** p_id, phone_no (multiple phones per patient)
3. **Dentist:** dentist_id, dentist_name
4. **Dentist Specialization:** dentist_id, specialization (multiple specializations)
5. **DentalProcedure:** pd_id, procedure_name, description
6. **Medication:** m_id, medication_name, dosage
7. **Tool:** serial_no (varchar), tool_name, end_of_life
8. **Visit:** v_id, visit_date, r_id (reminder reference)
9. **Reminder:** r_id, name, description
10. **Appointment:** ap_id, v_id, pd_id, m_id, serial_no
11. **VisitProcedure**: v_id, pd_id (link visit to procedure)

Multivalue Attributes Normalization:

Separate tables for multivalued attributes.
Patient Phone: Stores multiple phone numbers for each patient

Dentist Specialization: Stores multiple specializations per dentist

SQL Queries

- **CREATE**
- **SHOW**
- **DESCRIBE**
- **INSERT INTO**
- **UPDATE**
- **AS**
- **DISTINCT**
- **ORDER BY**
- **WHERE clause**
- **SELECT COMMAND**
- **ARITHMETIC OPERATORS**
- **RELATIONAL OPERATORS**
- **LOGICAL OPERATORS**
- **AND**
- **BETWEEN**
- **IN**
- **LIKE**
- **NULL and NOT NULL**
- **AGGREGATE FUNCTIONS**
- **GROUP BY**
- **HAVING**
- **DELETE**
- **DROP**
- **TRUNCATE**
- **JOINS**
- **GRANT PRIVILIGES**

Show DATABASES:

Show all databases and tables placed in a database MySQL use the following command:

show databases;

show tables;

CREATE:

It's used for the creation of tables or databases.

```
CREATE TABLE table_name (
   column1 datatype,
   column2 datatype,
   …);
```

DESCRIBE:

Shows the structure of a table.

```
DESCRIBE table_name;
```

INSERT:

Insert new records into a table.

```
INSERT INTO table_name (column1, column2, ...)
VALUES (value1, value2, ...);
```

UPDATE:

Modifies existing data.

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

AS

Gives a column or table an alias (temporary name).

```
SELECT column_name AS alias_name
FROM table_name;
```

DISTINCT

Removes duplicate records.

```
SELECT DISTINCT column_name
FROM table_name;
```

ORDER BY

Sorts results by one or more columns.

```
SELECT column1, column2
FROM table_name
ORDER BY column1 [ASC|DESC];
```

WHERE clause

Filters records.

```
SELECT column1, column2
FROM table_name
WHERE condition;
```

SELECT COMMAND

Retrieves data from one or more tables.

```
SELECT column1 + column2
FROM table_name;
```

RELATIONAL OPERATORS

Compare values (=, >, <, >=, <=, <>).

```
SELECT column1
FROM table_name
WHERE column2 [=|<>|>|<|>=|<=] value;
```

ARITHMETIC OPERATORS

Used in calculations (+, -, *, /, %).

```
SELECT column1 + column2
FROM table_name;
```

LOGICAL OPERATORS

Combine multiple conditions (AND, OR, NOT).

```
SELECT * FROM table_name
WHERE condition1 [AND|OR|NOT] condition2;
```

AND

Returns records that satisfy both conditions.

```
SELECT *
FROM table_name
WHERE condition1 AND condition2;
```

BETWEEN

Filters values within a specific range.

```
SELECT *
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

IN

Matches any value in a given list.

```
SELECT *
FROM table_name
```

```
WHERE column_name IN (value1, value2, ...);
```

LIKE

Searches for a pattern in a column (e.g., starts with 'A').

```
SELECT *
FROM table_name
WHERE column_name LIKE 'pattern';
```

NULL and NOT NULL

Checks for empty (NULL) or non-empty values.

```
SELECT *
FROM table_name
WHERE column_name IS NULL;
```

```
SELECT *
FROM table_name
WHERE column_name IS NOT NULL;
```

AGGREGATE FUNCTIONS

Performs calculations like:

```
SELECT AGGREGATE_FUNCTION(column_name)
FROM table_name;
```

- ✓ COUNT(): total records
- ✓ SUM(): total value
- ✓ AVG(): average
- ✓ MIN(): minimum
- ✓ MAX(): maximum

GROUP BY

Groups rows based on column values for aggregation.

```
SELECT column_name, AGGREGATE_FUNCTION(column_name)
FROM table_name
GROUP BY column_name;
```

HAVING

Filters grouped data (used with GROUP BY).

```
SELECT column_name, AGGREGATE_FUNCTION(column_name)
FROM table_name
GROUP BY column_name
HAVING condition;
```

Count & Count(*)

Used to count the number of rows in a table. This function counts all rows regardless of whether they contain NULL values.

select count (*) attribute_name from table;

select count (attribute_name) from table;

VIEWS

A MySQL view is a predefined select query that operates on existing data without duplicating it. A view acts as a virtual table.

DELETE

DELETE statement is used to delete rows in a table. It deletes a specific row using where clause.

DELETE from table where column_name= 'value';

DROP

DROP statement is used to delete the whole table along with table structure, attribute and indexes.

DROP table table_name;

TRUNCATE

TRUNCATE statement is used to delete all data in the table not the whole table.

TRUNCATE table_name;

JOINS

A JOIN is used to combine rows from two or more tables based on a related column between them.

TYPES OF JOINS:

1. INNER JOIN

Returns only the matching rows from both tables.

Syntax:

```
SELECT table1.column1, table2.column2
FROM table1
INNER JOIN table2
ON table1.common_column = table2.common_column;
```

2. LEFT JOIN

Returns all rows from the left table and matching rows from the right table. If no match, NULLs are returned for right table columns.

Syntax:

```
SELECT table1.column1, table2.column2
FROM table1
LEFT JOIN table2
ON table1.common_column = table2.common_column;
```

3. RIGHT JOIN

Returns all rows from the right table and matching rows from the left table. If no match, NULLs are returned for left table columns.

Syntax:

```
SELECT table1.column1, table2.column2
FROM table1
RIGHT JOIN table2
ON table1.common_column = table2.common_column;
```

4. Equi Join

An Equi Join is a type of INNER JOIN that uses the equality operator (=) to match rows from two or more tables based on a common column.

It is the most common form of join and retrieves rows with equal values

Syntax:

```
SELECT table1.column1, table2.column2
FROM table1, table2
WHERE table1.common_column = table2.common_column;
```

- **GRANT PRIVILEGES**

To grant privileges in MySQL, you use the GRANT statement. This allows a user to perform specific actions (like SELECT, INSERT, UPDATE, etc.) on a database or table.

Syntax:

GRANT privileges ON database.table TO 'username'@'host';

SUBQUERIES ON MySQL:

A subquery (or inner query) is a query nested within another SQL query. It is used to retrieve data that will be used in the main query (the outer query). Subqueries can be used in various clauses such as SELECT, FROM, WHERE, and HAVING. They allow for more complex queries by enabling the use of the result of one query as a condition for another.

Syntaxes:

Subquery in the SELECT Statement:

SELECT column1, column2, ...

FROM table_name

WHERE column_name IN

(SELECT column_name FROM table_name WHERE condition);

Subquery in the FROM Clause:

```
SELECT column1, column2, ...
FROM
(SELECT column1, column2 FROM table_name WHERE
condition)
AS alias_name;
```

Subquery in the WHERE Clause:

```
SELECT column1, column2, ...
FROM table_name
WHERE column_name = (SELECT column_name
FROM table_name WHERE condition);
```

SingLe and muLtipLe RoW:

- Single row returns the single value of the query.
- While on other hand the multiple row returns the more then the single row.

Practical implementation of queries

- **CREATE:**

```
mysql> CREATE database dental_clinic;
```

- **USE:**

```
mysql> use dental_clinic;
Database changed
```

- **DESCRIBE:**

```
mysql> desc visit;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| v_id              | int         | NO   | PRI | NULL    |       |
| visitDate         | date        | NO   |     | NULL    |       |
| reason            | varchar(60) | YES  |     | NULL    |       |
| Pain_level        | int         | YES  |     | NULL    |       |
| FollowUp_visit_id | int         | YES  | MUL | NULL    |       |
| p_id              | int         | NO   | MUL | NULL    |       |
| d_id              | int         | NO   | MUL | NULL    |       |
| t_id              | int         | NO   | MUL | NULL    |       |
| r_id              | int         | YES  | MUL | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
9 rows in set (0.03 sec)
```

- **UPDATE**

```
mysql> UPDATE treatment SET den_id='4' Where t_id='678';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE treatment SET den_id='5' Where t_id='680';
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE treatment SET den_id='6' Where t_id='681';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from treatment;
+------+--------------------+-----------------------------------+--------+--------+
| t_id | name               | description                       | pat_id | den_id |
+------+--------------------+-----------------------------------+--------+--------+
|  678 | Teeth Cleaning     | Professional removal of plaque    |    101 |      4 |
|  680 | Tooth extraction   | Removal of a damaged,decayed tooth|    102 |      5 |
|  681 | Braces Installation| Fitting of dental braces          |    103 |      6 |
+------+--------------------+-----------------------------------+--------+--------+
3 rows in set (0.00 sec)
```

- **AS**

```
mysql> SELECT p_id AS Patient FROM patient;
+---------+
| Patient |
+---------+
|       1 |
|       2 |
|       3 |
+---------+
3 rows in set (0.00 sec)
```

- **DISTINCT**

```
mysql> select DISTINCT dep_no from employee;
+--------+
| dep_no |
+--------+
|     11 |
|     12 |
|     13 |
```

- **ORDER BY**

```
mysql> select JOB ,e_name from employee;
+----------+--------+
| JOB      | e_name |
+----------+--------+
| SI       | Arhum  |
| CONSTBLE | Murat  |
| COMMANDO | Faiza  |
| SI       | Fazan  |
+----------+--------+
4 rows in set (0.00 sec)

mysql> select e_name from employee ORDER BY salary DESC;
+--------+
| e_name |
+--------+
| Faiza  |
| Murat  |
| Fazan  |
| Arhum  |
+--------+
4 rows in set (0.00 sec)
```

- **WHERE clause**

```
mysql> UPDATE visit SET r_id = 33 where v_id =93;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

- **SELECT COMMAND**

```
mysql> select dep_no from department where (dep_no!=12);
+--------+
| dep_no |
+--------+
|     11 |
|     13 |
|     14 |
+--------+
3 rows in set (0.00 sec)

mysql>
```

```
mysql> select * from medication;
+------+-------------+--------+
| m_id | Name        | Dosage |
+------+-------------+--------+
|   11 | Ibuprofen   | 200 mg |
|   12 | Amoxicillin | 500 mg |
|   13 | Paracetamol | 500 mg |
+------+-------------+--------+
3 rows in set (0.00 sec)
```

- **ARITHMETIC OPERATORS**

```
mysql> select salary+5000 as salary_bonus from dentist where d_id='4';
+--------------+
| salary_bonus |
+--------------+
|        55000 |
+--------------+
1 row in set (0.00 sec)

mysql> select salary-6000 as salary_deduct from dentist where d_id='5';
+---------------+
| salary_deduct |
+---------------+
|         24000 |
+---------------+
1 row in set (0.00 sec)

mysql> select salary+500*12 as salary_bonus from dentist where d_id='4';
+--------------+
| salary_bonus |
+--------------+
|        56000 |
+--------------+
1 row in set (0.00 sec)
```

- **RELATIONAL OPERATORS**

```
mysql> select d_name from dentist where salary>50000;
+-----------+
| d_name    |
+-----------+
| Dr.Maria  |
+-----------+
1 row in set (0.00 sec)

mysql> select d_anme from dentist where salary=50000;
ERROR 1054 (42S22): Unknown column 'd_anme' in 'field list'
mysql>
mysql> select d_name from dentist where salary=50000;
+----------+
| d_name   |
+----------+
| Dr.Ijaz  |
+----------+
1 row in set (0.00 sec)

mysql> select d_name from dentist where salary<50000;
+------------+
| d_name     |
+------------+
| Dr.Hassan  |
+------------+
1 row in set (0.00 sec)
```

- **LOGICAL OPERATORS**

```
mysql> select description from treatment WHERE pat_id='102' AND t_id='680';
+-----------------------------------+
| description                       |
+-----------------------------------+
| Removal of a damaged,decayed tooth |
+-----------------------------------+
1 row in set (0.00 sec)

mysql> select description from treatment WHERE pat_id='102' or t_id='681';
+-----------------------------------+
| description                       |
+-----------------------------------+
| Removal of a damaged,decayed tooth |
| Fitting of dental braces          |
+-----------------------------------+
2 rows in set (0.02 sec)

mysql>
```

- **AND & BETWEEN**

```
mysql> select * from dentist WHERE (d_id=2 AND salary BETWEEN 1000 AND 30000);
+------+-----------+----------+--------+
| d_id | FirstName | LastName | salary |
+------+-----------+----------+--------+
|    2 | Farukh    | Awais    |  17837 |
+------+-----------+----------+--------+
1 row in set (0.00 sec)
```

- **IN**

```
mysql> SELECT * FROM patient where FirstName IN("Arhum","Murat");
+------+-----------+----------+------------+
| p_id | FirstName | LastName | DOB        |
+------+-----------+----------+------------+
|    1 | Arhum     | Rana     | 2001-05-19 |
|    3 | Murat     | Ansari   | 2003-12-05 |
+------+-----------+----------+------------+
2 rows in set (0.00 sec)
```

- **LIKE**

```
mysql> SELECT * FROM patient where FirstName IN("Arhum","Murat");
+------+-----------+----------+------------+
| p_id | FirstName | LastName | DOB        |
+------+-----------+----------+------------+
|    1 | Arhum     | Rana     | 2001-05-19 |
|    3 | Murat     | Ansari   | 2003-12-05 |
+------+-----------+----------+------------+
2 rows in set (0.00 sec)

mysql> select FirstName ,LastName FROM patient WHERE FirstName LIKE 'A%';
+-----------+----------+
| FirstName | LastName |
+-----------+----------+
| Arhum     | Rana     |
+-----------+----------+
1 row in set (0.01 sec)

mysql> select FirstName ,LastName FROM patient WHERE FirstName LIKE '_U%';
+-----------+----------+
| FirstName | LastName |
+-----------+----------+
| Murat     | Ansari   |
+-----------+----------+
1 row in set (0.00 sec)

mysql> SELECT LastName FROM patient WHERE FirstName LIKE '_%_i%';
+----------+
| LastName |
+----------+
| Rahim    |
+----------+
1 row in set (0.00 sec)
```

- **IS NULL and IS NOT NULL**

```
mysql> SELECT * FROM visit WHERE FollowUp_visit_id IS NULL;
+------+-----------+-----------------------+-----------+-------------------+------+------+------+------+
| v_id | visitDate | reason                | Pain_level | FollowUp_visit_id | p_id | d_id | t_id | r_id |
+------+-----------+-----------------------+-----------+-------------------+------+------+------+------+
|   91 | 2004-09-29 | Routine dental checkUp |         4 |              NULL |    1 |    2 |  101 |   31 |
|   92 | 2001-12-29 | ROOT CANAL             |         7 |              NULL |    3 |    3 |  201 |   32 |
+------+-----------+-----------------------+-----------+-------------------+------+------+------+------+
2 rows in set (0.01 sec)

mysql> SELECT v_id , visitDate, Pain_level FROM visit WHERE p_id IS NOT NULL;
+------+-----------+------------+
| v_id | visitDate | Pain_level |
+------+-----------+------------+
|   91 | 2004-09-29 |          4 |
|   92 | 2001-12-29 |          7 |
|   93 | 2002-01-19 |          4 |
+------+-----------+------------+
3 rows in set (0.00 sec)
```

- **AGGREGATE FUNCTIONS**

```
mysql> SELECT COUNT(*)  AS TOTAL FROM visit ;
+--------+
| TOTAL  |
+--------+
|      3 |
+--------+
1 row in set (0.01 sec)

mysql> SELECT COUNT(FollowUp_visit_id) AS the VISITOR FROM visit;
ERROR 1064 (42000): You have an error in your SQL syntax; check the
t' at line 1
mysql> SELECT COUNT(FollowUp_visit_id) AS VISITORS FROM visit;
+-----------+
| VISITORS  |
+-----------+
|         1 |
+-----------+
1 row in set (0.01 sec)

mysql> SELECT AVG(r_id) FROM reminder;
+------------+
| AVG(r_id)  |
+------------+
|   32.0000  |
+------------+
```

```
mysql> SELECT MAX(salary) FROM dentist Where d_id=2;
+-------------+
| MAX(salary) |
+-------------+
|       17837 |
+-------------+
1 row in set (0.01 sec)

mysql> SELECT MAX(salary) FROM dentist;
+-------------+
| MAX(salary) |
+-------------+
|       17837 |
+-------------+
1 row in set (0.00 sec)
```

```
mysql> select MIN(salary) FROM dentist;
+-------------+
| MIN(salary) |
+-------------+
|       16385 |
+-------------+
1 row in set (0.00 sec)

mysql> select SUM (salary) FROM dentist;
ERROR 1630 (42000): FUNCTION dental_clinic.SUM does not exi
mysql> select SUM(salary) FROM dentist;
+-------------+
| SUM(salary) |
+-------------+
|       52059 |
+-------------+
1 row in set (0.00 sec)
```

- **GROUP BY**

```
mysql> select AVG(salary) FROM dentist GROUP BY FirstName;
+-------------+
| AVG(salary) |
+-------------+
|  17837.0000 |
|  17837.0000 |
|  16385.0000 |
+-------------+
3 rows in set (0.01 sec)
```

- **HAVING**

```
mysql> select COUNT(*) FROM dentist GROUP BY FirstName having COUNT(d_id) IS NOT NULL;
+----------+
| COUNT(*) |
+----------+
|        1 |
|        1 |
|        1 |
+----------+
3 rows in set (0.00 sec)
```

- **INNER:**

```
mysql> select * from patient INNER JOIN treatment ON patient.p_id=treatment.pat_id;
+------+--------+------------+------+-------------------+---------------------------------+--------+
| p_id | p_name | dob        | t_id | name              | description                     | pat_id |
+------+--------+------------+------+-------------------+---------------------------------+--------+
|  101 | Sara   | 2002-11-10 |  678 | Teeth Cleaning    | Professional removal of plaque  |    101 |
|  102 | Zainab | 2005-01-04 |  680 | Tooth extraction  | Removal of a damaged,decayed tooth |    102 |
|  103 | Ahmad  | 2006-09-14 |  681 | Braces Installation | Fitting of dental braces      |    103 |
+------+--------+------------+------+-------------------+---------------------------------+--------+
```

- **RIGHT & LEFT JOIN:**

```
mysql> select * from patient RIGHT JOIN treatment ON patient.p_id=treatment.pat_id;
+------+--------+------------+------+---------------------+---------------------------------+--------+
| p_id | p_name | dob        | t_id | name                | description                     | pat_id |
+------+--------+------------+------+---------------------+---------------------------------+--------+
|  101 | Sara   | 2002-11-10 |  678 | Teeth Cleaning      | Professional removal of plaque  |    101 |
|  102 | Zainab | 2005-01-04 |  680 | Tooth extraction    | Removal of a damaged,decayed tooth |    102 |
|  103 | Ahmad  | 2006-09-14 |  681 | Braces Installation | Fitting of dental braces        |    103 |
+------+--------+------------+------+---------------------+---------------------------------+--------+
3 rows in set (0.00 sec)

mysql> select * from patient LEFT JOIN treatment ON patient.p_id=treatment.pat_id;
+------+--------+------------+------+---------------------+---------------------------------+--------+
| p_id | p_name | dob        | t_id | name                | description                     | pat_id |
+------+--------+------------+------+---------------------+---------------------------------+--------+
|  101 | Sara   | 2002-11-10 |  678 | Teeth Cleaning      | Professional removal of plaque  |    101 |
|  102 | Zainab | 2005-01-04 |  680 | Tooth extraction    | Removal of a damaged,decayed tooth |    102 |
|  103 | Ahmad  | 2006-09-14 |  681 | Braces Installation | Fitting of dental braces        |    103 |
+------+--------+------------+------+---------------------+---------------------------------+--------+
3 rows in set (0.00 sec)
```

- **Combining THREE tables together using JOINS:**

```
mysql> select * from treatment;
+------+--------------------+---------------------------------+--------+--------+
| t_id | name               | description                     | pat_id | den_id |
+------+--------------------+---------------------------------+--------+--------+
|  678 | Teeth Cleaning     | Professional removal of plaque  |    101 |      4 |
|  680 | Tooth extraction   | Removal of a damaged,decayed tooth |    102 |      5 |
|  681 | Braces Installation | Fitting of dental braces        |    103 |      6 |
+------+--------------------+---------------------------------+--------+--------+
3 rows in set (0.00 sec)

mysql> select patient.p_name,dentist.d_name,treatment.name,treatment.description FROM treatment INNER JOIN patient ON treatment.pat_id
=patient.p_id INNER JOIN dentist ON treatment.den_id=dentist.d_id;
+--------+-----------+--------------------+---------------------------------+
| p_name | d_name    | name               | description                     |
+--------+-----------+--------------------+---------------------------------+
| Sara   | Dr.Ijaz   | Teeth Cleaning     | Professional removal of plaque  |
| Zainab | Dr.Hassan | Tooth extraction   | Removal of a damaged,decayed tooth |
| Ahmad  | Dr.Maria  | Braces Installation | Fitting of dental braces        |
+--------+-----------+--------------------+---------------------------------+
3 rows in set (0.00 sec)

mysql>
```

GRANT USER:

```
Database changed
mysql> CREATE user 'patient'@'localhost';
Query OK, 0 rows affected (0.07 sec)

mysql> show Grants for 'patient'@'localhost';
+-------------------------------------------------+
| Grants for patient@localhost                    |
+-------------------------------------------------+
| GRANT USAGE ON *.* TO `patient`@`localhost`     |
+-------------------------------------------------+
1 row in set (0.00 sec)
```

SHOW ALL PRIVILEGES;

```
mysql> show privileges;
+---------------------------+---------------------------------+-------------------------------------------------------+
| Privilege                 | Context                         | Comment                                               |
+---------------------------+---------------------------------+-------------------------------------------------------+
| Alter                     | Tables                          | To alter the table                                    |
| Alter routine             | Functions,Procedures            | To alter or drop stored functions/procedures          |
| Create                    | Databases,Tables,Indexes        | To create new databases and tables                    |
| Create routine            | Databases                       | To use CREATE FUNCTION/PROCEDURE                      |
| Create role               | Server Admin                    | To create new roles                                   |
| Create temporary tables   | Databases                       | To use CREATE TEMPORARY TABLE                         |
| Create view               | Tables                          | To create new views                                   |
| Create user               | Server Admin                    | To create new users                                   |
| Delete                    | Tables                          | To delete existing rows                               |
| Drop                      | Databases,Tables                | To drop databases, tables, and views                  |
| Drop role                 | Server Admin                    | To drop roles                                         |
| Event                     | Server Admin                    | To create, alter, drop and execute events             |
| Execute                   | Functions,Procedures            | To execute stored routines                            |
| File                      | File access on server           | To read and write files on the server                 |
| Grant option              | Databases,Tables,Functions,Procedures | To give to other users those privileges you possess |
| Index                     | Tables                          | To create or drop indexes                             |
| Insert                    | Tables                          | To insert data into tables                            |
| Lock tables               | Databases                       | To use LOCK TABLES (together with SELECT privilege)   |
| Process                   | Server Admin                    | To view the plain text of currently executing queries |
| Proxy                     | Server Admin                    | To make proxy user possible                           |
| References                | Databases,Tables                | To have references on tables                          |
| Reload                    | Server Admin                    | To reload or refresh tables, logs and privileges      |
| Replication client        | Server Admin                    | To ask where the slave or master servers are          |
| Replication slave         | Server Admin                    | To read binary log events from the master             |
| Select                    | Tables                          | To retrieve rows from table                           |
| Show databases            | Server Admin                    | To see all databases with SHOW DATABASES              |
| Show view                 | Tables                          | To see views with SHOW CREATE VIEW                    |
| Shutdown                  | Server Admin                    | To shut down the server                               |
| Super                     | Server Admin                    | To use KILL thread, SET GLOBAL, CHANGE MASTER, etc.   |
| Trigger                   | Tables                          | To use triggers                                       |
| Create tablespace         | Server Admin                    | To create/alter/drop tablespaces                      |
| Update                    | Tables                          | To update existing rows                               |
| Usage                     | Server Admin                    | No privileges - allow connect only                    |
| ENCRYPTION_KEY_ADMIN      | Server Admin                    |                                                       |
```

SingLe ROW and Multi ROW SUBQueries

```
mysql> SELECT FirstName FROM patient  WHERE p_id IN(Select p_id FROM visit WHERE pain_level>5);
+-----------+
| FirstName |
+-----------+
| Murat     |
+-----------+
1 row in set (0.00 sec)

mysql> SELECT * FROM medication WHERE m_id IN(SELECT m_id FROM appointment WHERE a_id!=0);
+------+------------+--------+
| m_id | Name       | Dosage |
+------+------------+--------+
|   11 | Ibuprofen  | 200 mg |
|   12 | Amoxicillin | 500 mg |
|   13 | Paracetamol | 500 mg |
+------+------------+--------+
3 rows in set (0.00 sec)
```

DEFAULT CONSTAINTS:

```
mysql> ALTER table patient ADD AGE INT(255) DEFAULT 18;
Query OK, 0 rows affected, 1 warning (0.42 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> select * from patient;
+------+-----------+----------+------------+------+
| p_id | FirstName | LastName | DOB        | AGE  |
+------+-----------+----------+------------+------+
|    1 | Arhum     | Rana     | 2001-05-19 |   18 |
|    2 | Yasir     | Rahim    | 2004-02-03 |   18 |
|    3 | Murat     | Ansari   | 2003-12-05 |   18 |
+------+-----------+----------+------------+------+
3 rows in set (0.00 sec)
```

Check CONSTRAINTS:

```
mysql> ALTER TABLE patient ADD Gender CHAR(1) CHECK (Gender IN ('M', 'F'));
```

Select * from patient;

```
mysql> select * from patient;
+------+-----------+----------+------------+------+--------+
| p_id | FirstName | LastName | DOB        | AGE  | Gender |
+------+-----------+----------+------------+------+--------+
|    1 | Arhum     | Rana     | 2001-05-19 |   18 | F      |
|    2 | Yasir     | Rahim    | 2004-02-03 |   18 | M      |
|    3 | Murat     | Ansari   | 2003-12-05 |   18 | M      |
+------+-----------+----------+------------+------+--------+
3 rows in set (0.00 sec)
```

View:

```
mysql> CREATE VIEW patient_dentist_view AS SELECT p.p_id, p.DOB FROM patient p JOIN denstist d ON p.p_id!=d_id;
ERROR 1146 (42S02): Table 'dental_clinic.denstist' doesn't exist
mysql> CREATE VIEW patient_dentist_view AS SELECT p.p_id, p.DOB FROM patient p JOIN dentist d ON p.p_id!=d_id;
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT * FROM patient_dentist_view
    -> ;
+------+------------+
| p_id | DOB        |
+------+------------+
|    3 | 2003-12-05 |
|    2 | 2004-02-03 |
|    3 | 2003-12-05 |
|    1 | 2001-05-19 |
|    2 | 2004-02-03 |
|    1 | 2001-05-19 |
+------+------------+
6 rows in set (0.01 sec)
```