NORMALIZATION & TABLES

GROUP#14

NEELAM SHEHZADI(110852)

SHANZAY MAHMOOD(110828

Submitted to :

Prof. Sherish Khan

QUERIES USED FOR MAKING TABLES

```
CREATE DATABASE database_name;

USE database_name;

CREATE TABLE table_name (    column_name1 DATA_TYPE,    column_name2 DATA_TYPE,…...);

INSERT INTO table_name (column_name1, column_name2, ...)VALUES (value1, value2,…...);

SHOW DATABASES;

SHOW TABLES;

DESCRIBE table_name;

UPDATE table_name SET column_name = new_valueWHERE condition;
```

QUERIES USED FOR MAKING TABLES

ALTER TABLE table_name ADD COLUMN column_name DATA_TYPE;

ALTER TABLE table_name MODIFY COLUMN column_name NEW_DATA_TYPE;

ALTER TABLE table name DROP COLUMN column_name;

DELETE FROM table_name WHERE condition;

DROP TABLE table_name;

DROP DATABASE database_name;

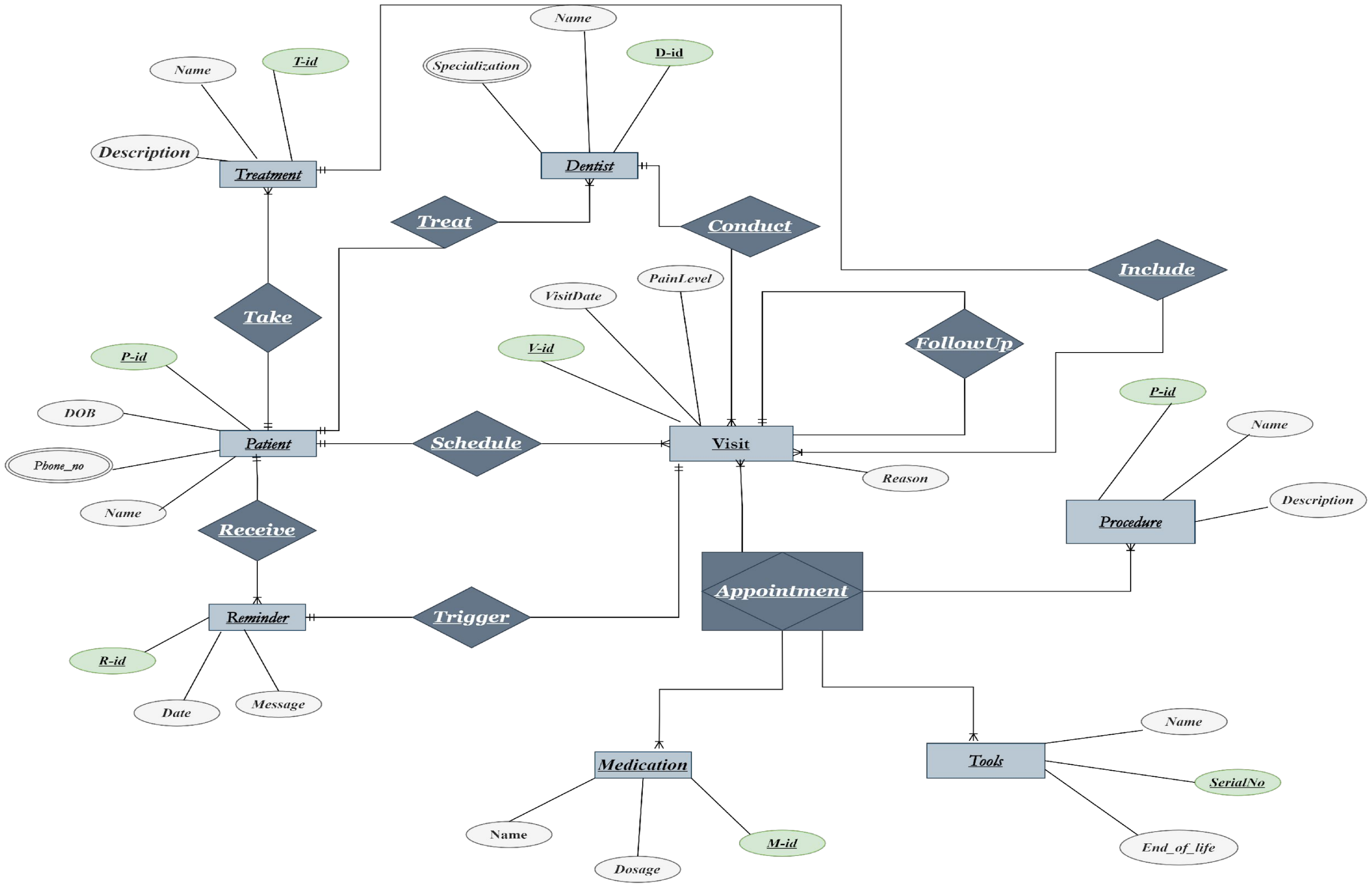
SELECT * FROM table_name;

SELECT column_name1, column_name2 FROM table_name;

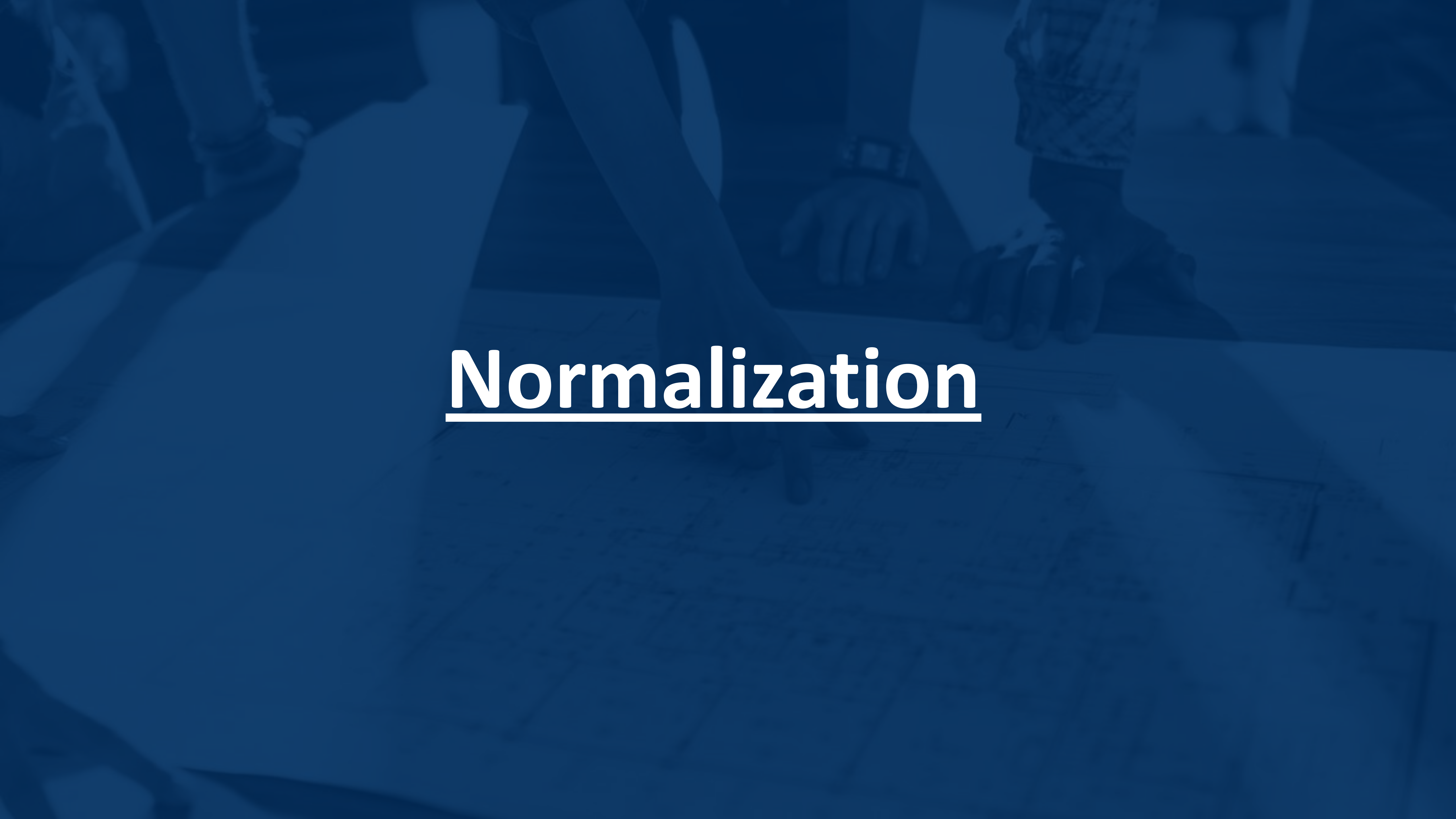
SELECT * FROM table_nameWHERE condition;

SELECT column_name AS alias_name FROM table_name;

ENTITY RELATIONSHIP DIAGRAM



Dental Clinic Database



Normalization

- 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.

MYSQL TABLES AND ENTITIES:

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

Patient: p_id, patient_name

Patient Phone: p_id, phone_no (multiple phones per patient)

Dentist: dentist_id, dentist_name

Dentist Specialization: dentist_id, specialization (multiple specializations)

DentalProcedure: pd_id, procedure_name, description

Medication: m_id, medication_name, dosage

Tool: serial_no (varchar), tool_name, end_of_life

Visit: v_id, visit_date, r_id (reminder reference)

Reminder: r_id, name, description

Appointment: ap_id, v_id, pd_id, m_id, serial_no

VisitProcedure: v_id, pd_id (link visit to procedure)

Multivalue Attributes Normalization:

Since relational databases do not support multivalue attributes directly, we create separate tables:

Patient Phone: Stores multiple phone numbers for each patient

Dentist Specialization: Stores multiple specializations per dentist

TABLES

OVERVIEW

ENTITIES:
With their attributes.

ENTITIES
Becomes tables while when you are dealing with the relation schema.

TABLES
Of each entites along with select command.

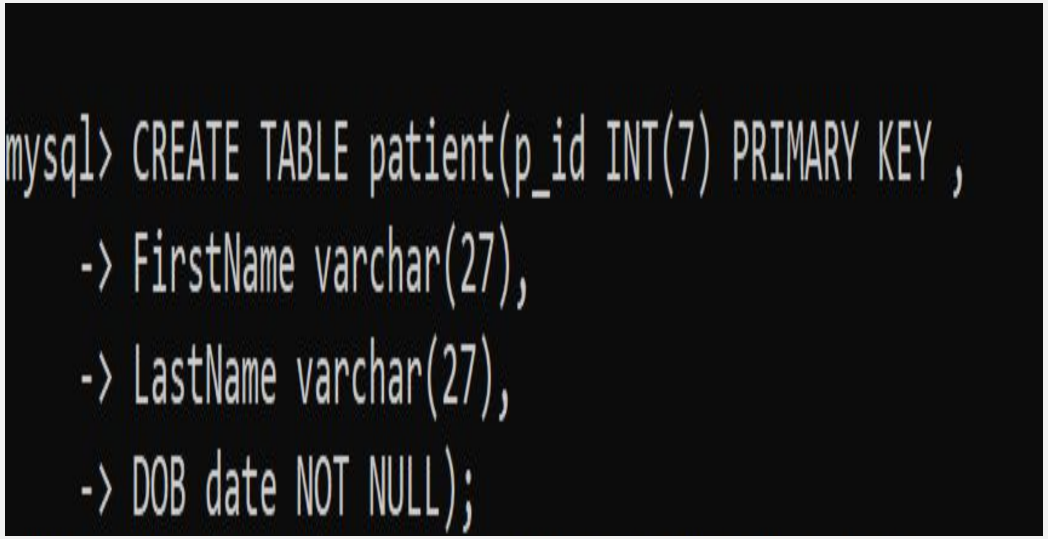
INSERT
Command on MYSQL.

UPDATE
on MySql.

AS
Alias on MySql.

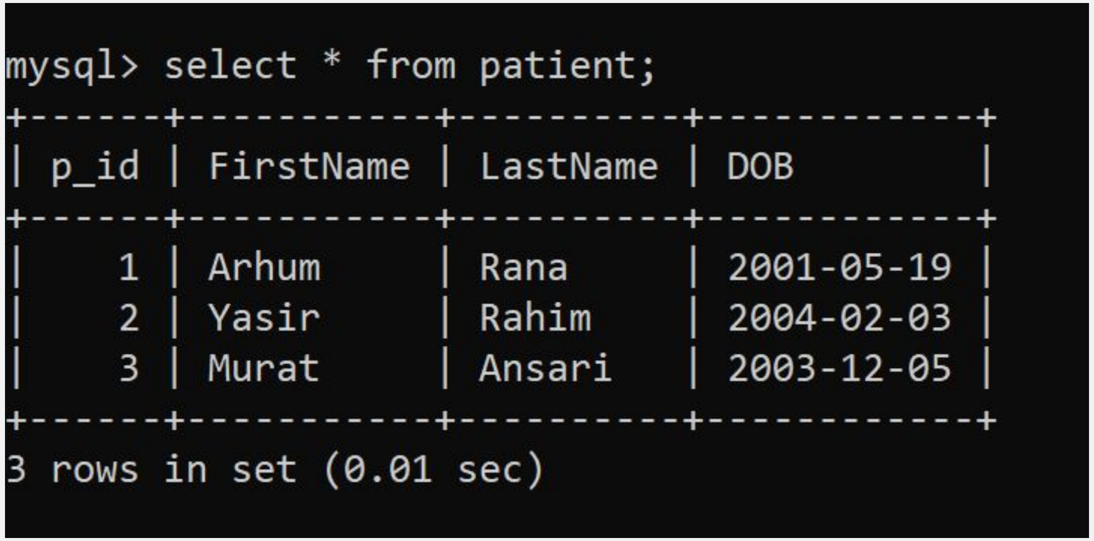
PATIENT TABLE:

Query on MYSQL:



```
mysql> CREATE TABLE patient(p_id INT(7) PRIMARY KEY ,
    -> FirstName varchar(27),
    -> LastName varchar(27),
    -> DOB date NOT NULL);
```

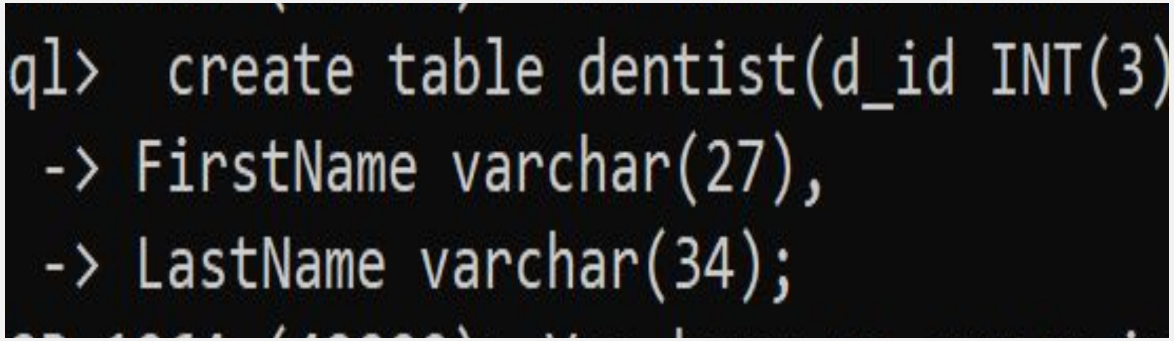
Patient table containing values:



```
mysql> select * from patient;
+------+-----------+----------+------------+
| p_id | FirstName | LastName | DOB        |
+------+-----------+----------+------------+
|    1 | Arhum     | Rana     | 2001-05-19 |
|    2 | Yasir     | Rahim    | 2004-02-03 |
|    3 | Murat     | Ansari   | 2003-12-05 |
+------+-----------+----------+------------+
3 rows in set (0.01 sec)
```

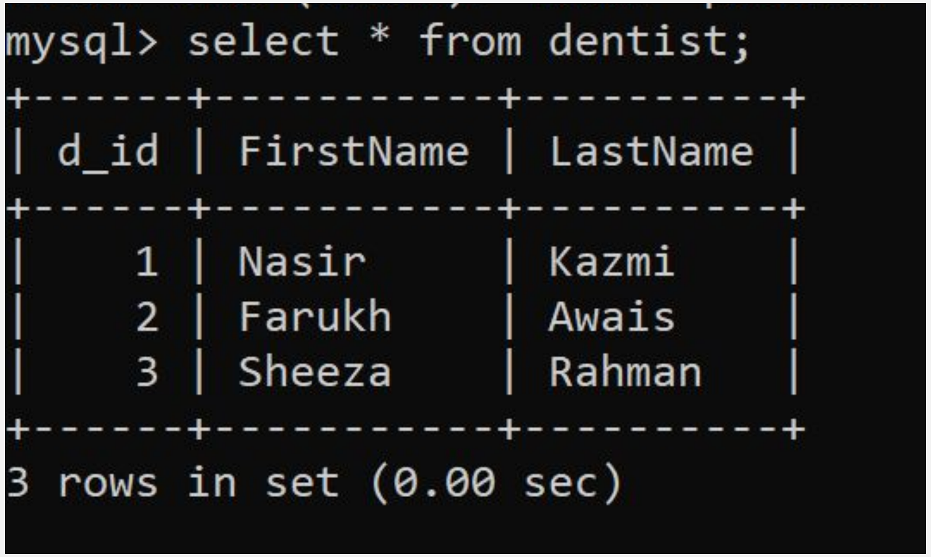
DENTIST TABLE:

Query to create table:



```
ql>  create table dentist(d_id INT(3)
 -> FirstName varchar(27),
 -> LastName varchar(34);
```

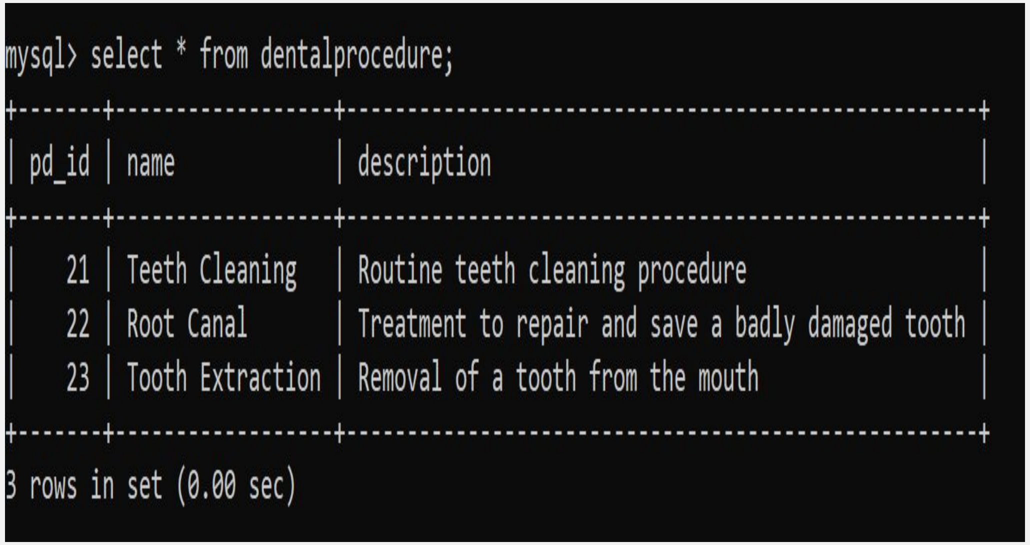
Dentist table along attributes:



```
mysql> select * from dentist;
+------+-----------+----------+
| d_id | FirstName | LastName |
+------+-----------+----------+
|    1 | Nasir     | Kazmi    |
|    2 | Farukh    | Awais    |
|    3 | Sheeza    | Rahman   |
+------+-----------+----------+
3 rows in set (0.00 sec)
```

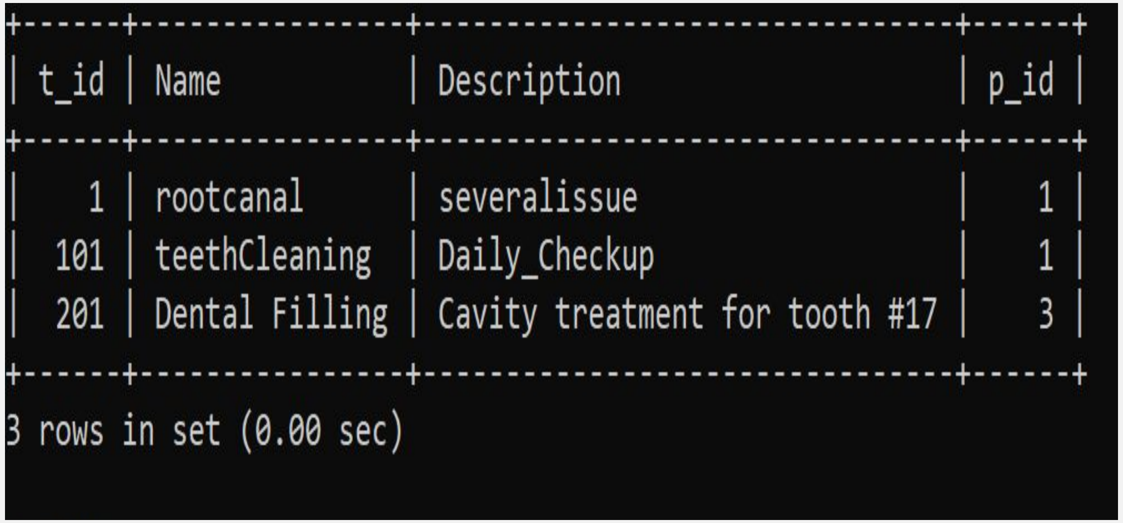
TREATMENT AND PROCEDURE TABLES:

Procedure table:



```
mysql> select * from dentalprocedure;
+-------+------------------+----------------------------------------------------+
| pd_id | name             | description                                        |
+-------+------------------+----------------------------------------------------+
|    21 | Teeth Cleaning   | Routine teeth cleaning procedure                   |
|    22 | Root Canal       | Treatment to repair and save a badly damaged tooth |
|    23 | Tooth Extraction | Removal of a tooth from the mouth                  |
+-------+------------------+----------------------------------------------------+
3 rows in set (0.00 sec)
```

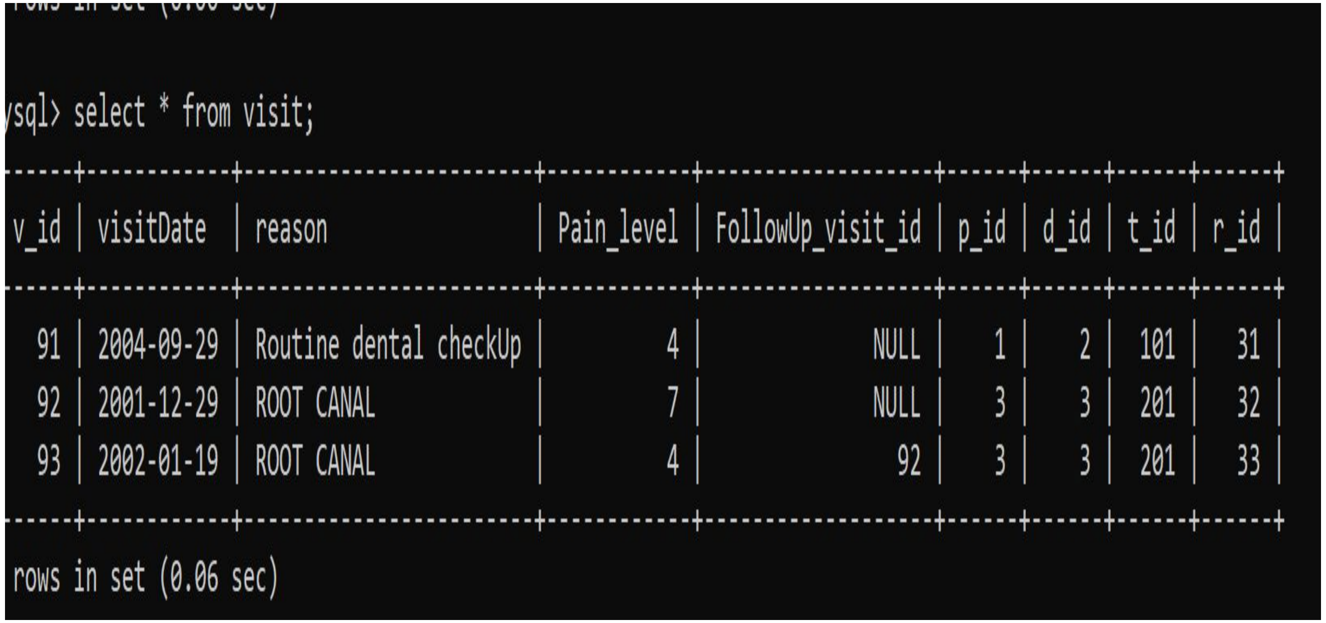
Treatment table along attributes:



```
+------+----------------+--------------------------------+------+
| t_id | Name           | Description                    | p_id |
+------+----------------+--------------------------------+------+
|    1 | rootcanal      | severalissue                   |    1 |
|  101 | teethCleaning  | Daily_Checkup                  |    1 |
|  201 | Dental Filling | Cavity treatment for tooth #17 |    3 |
+------+----------------+--------------------------------+------+
3 rows in set (0.00 sec)
```

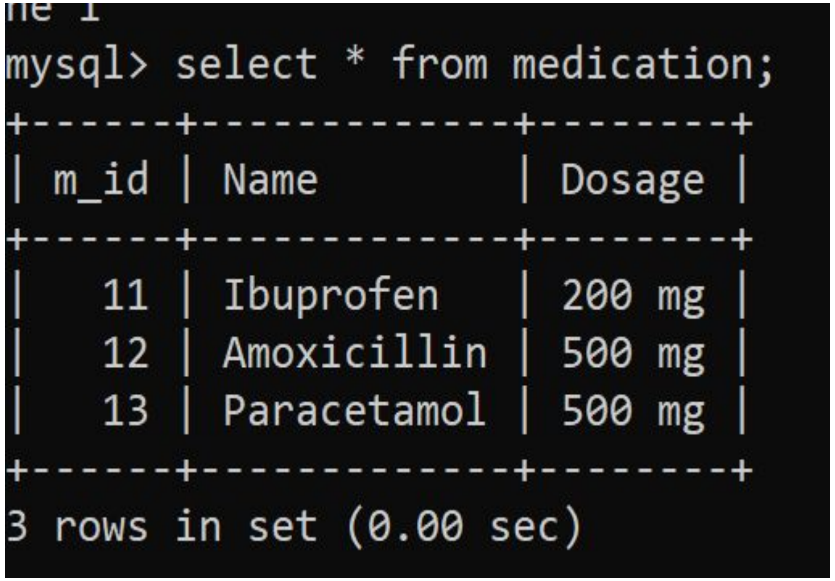
VISIT AND MEDICATION TABLES:

Visit table with attributes:



```
rows in set (0.00 sec)

ysql> select * from visit;
-----+------------+----------------------+------------+------------------+------+------+------+------+
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 |
  93 | 2002-01-19 | ROOT CANAL           |          4 |               92 |    3 |    3 |  201 |   33 |
-----+------------+----------------------+------------+------------------+------+------+------+------+
rows in set (0.06 sec)
```

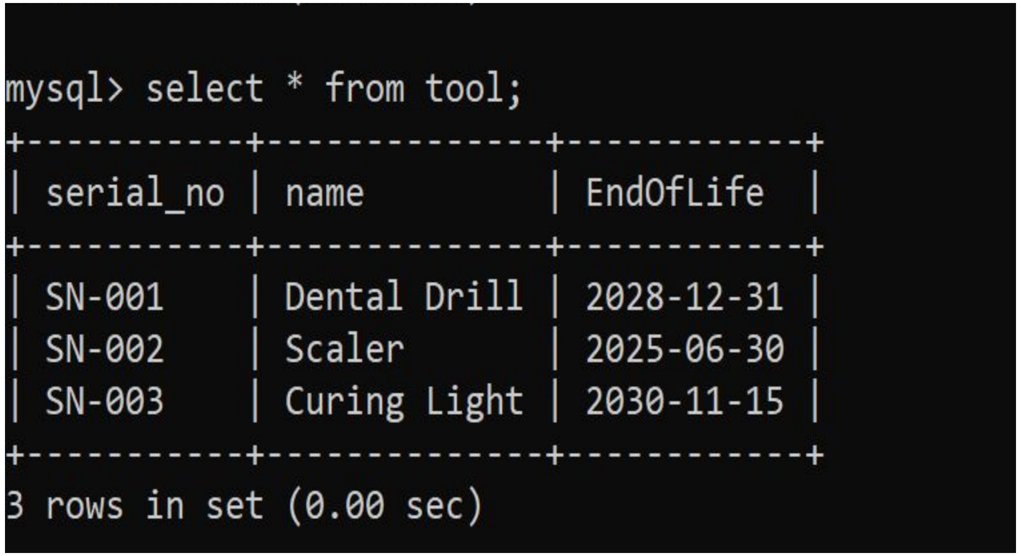
Medication:



```
ne 1
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)
```

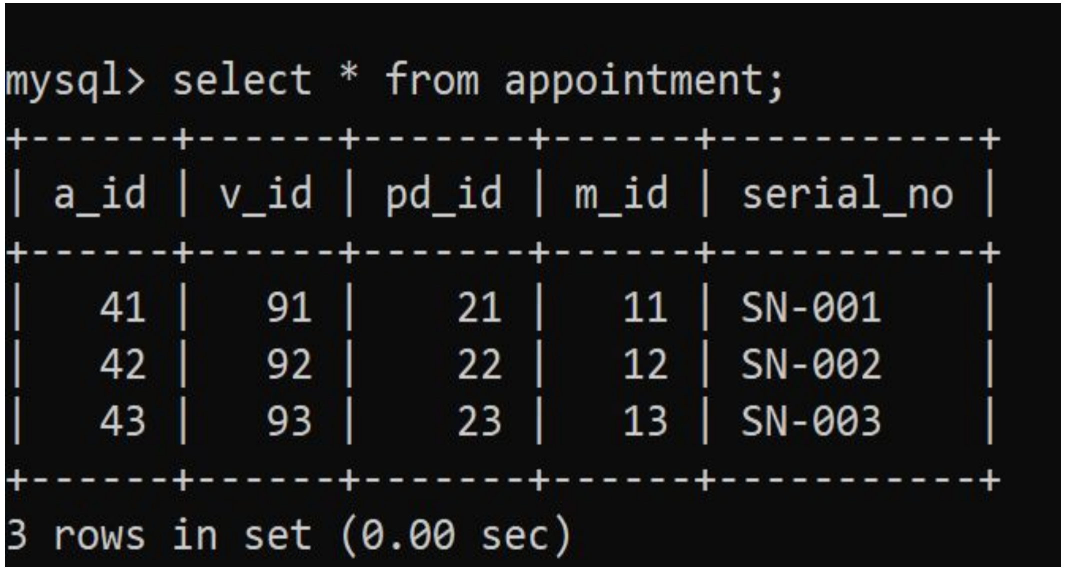
APPOINTMENT AND TOOL:

TOOl\Equipments:



```
mysql> select * from tool;
+-----------+--------------+------------+
| serial_no | name         | EndOfLife  |
+-----------+--------------+------------+
| SN-001    | Dental Drill | 2028-12-31 |
| SN-002    | Scaler       | 2025-06-30 |
| SN-003    | Curing Light | 2030-11-15 |
+-----------+--------------+------------+
3 rows in set (0.00 sec)
```

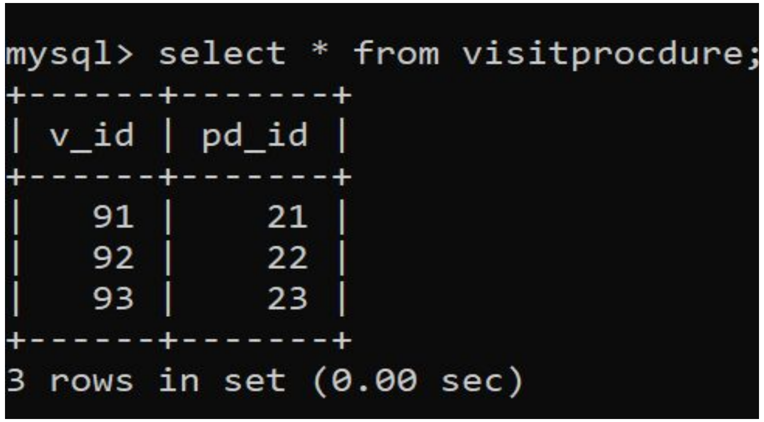
Appointment(Associative):



```
mysql> select * from appointment;
+------+------+-------+------+-----------+
| a_id | v_id | pd_id | m_id | serial_no |
+------+------+-------+------+-----------+
|   41 |   91 |    21 |   11 | SN-001    |
|   42 |   92 |    22 |   12 | SN-002    |
|   43 |   93 |    23 |   13 | SN-003    |
+------+------+-------+------+-----------+
3 rows in set (0.00 sec)
```

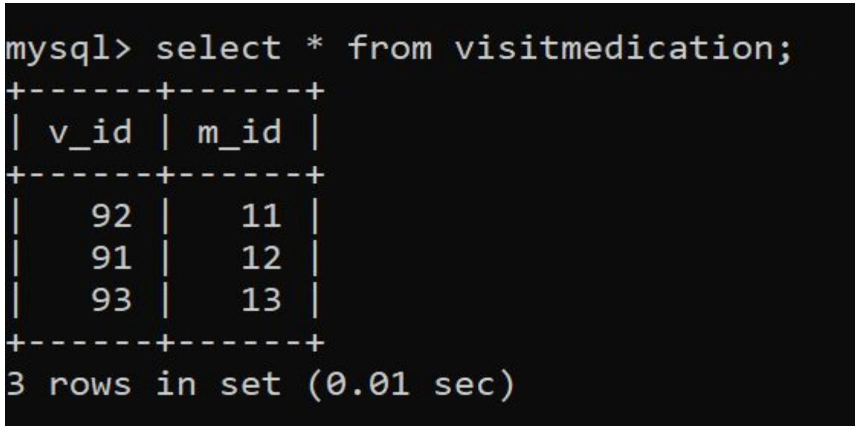
JUNCTION TABLES FOR M:M RELATION:

VisitProcedure:



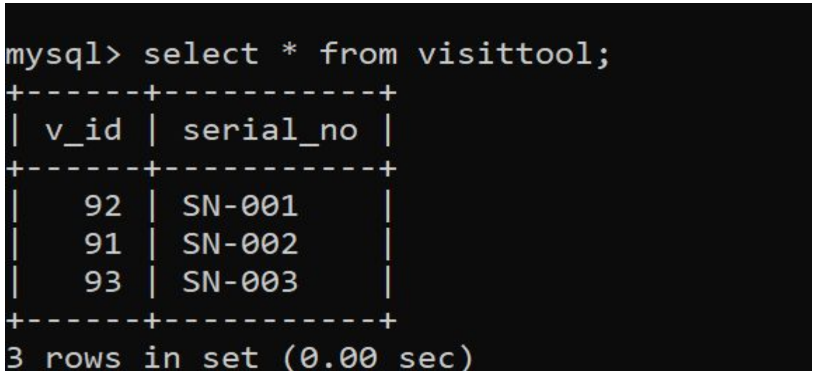
```
mysql> select * from visitprocdure;
+------+-------+
| v_id | pd_id |
+------+-------+
|   91 |    21 |
|   92 |    22 |
|   93 |    23 |
+------+-------+
3 rows in set (0.00 sec)
```

VisitMedication:



```
mysql> select * from visitmedication;
+------+------+
| v_id | m_id |
+------+------+
|   92 |   11 |
|   91 |   12 |
|   93 |   13 |
+------+------+
3 rows in set (0.01 sec)
```

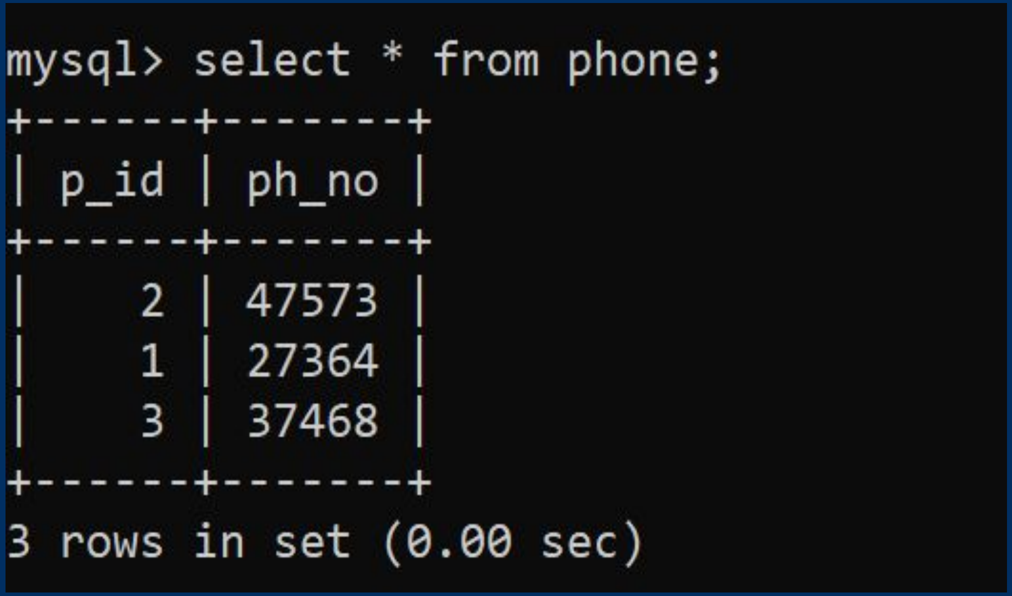
VisitTool:



```
mysql> select * from visittool;
+------+-----------+
| v_id | serial_no |
+------+-----------+
|   92 | SN-001    |
|   91 | SN-002    |
|   93 | SN-003    |
+------+-----------+
3 rows in set (0.00 sec)
```

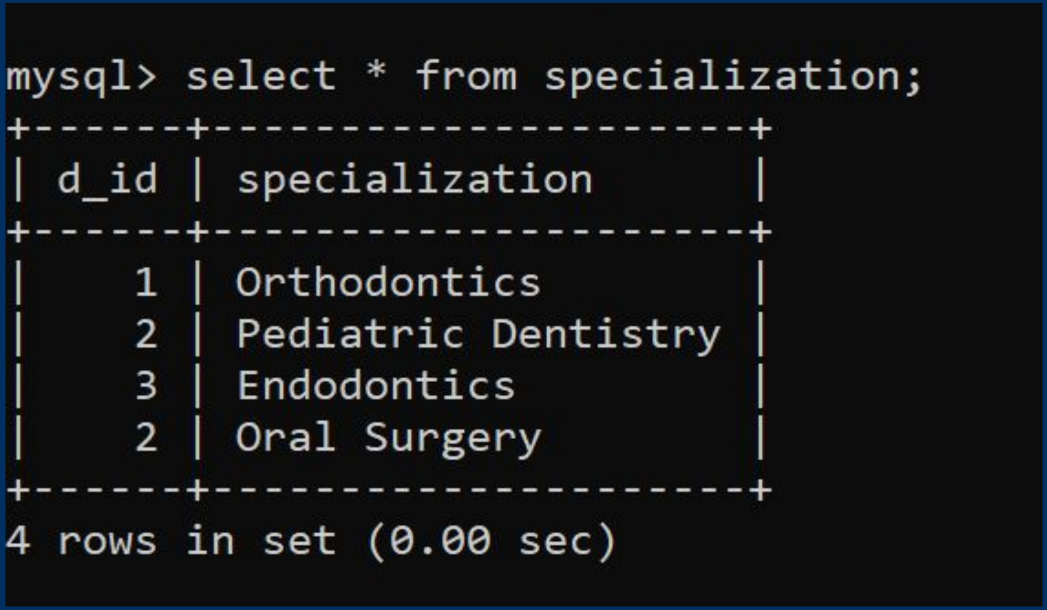
MULTIVALUE ATTRIBUTES:

PHONEN_NO in Patient:



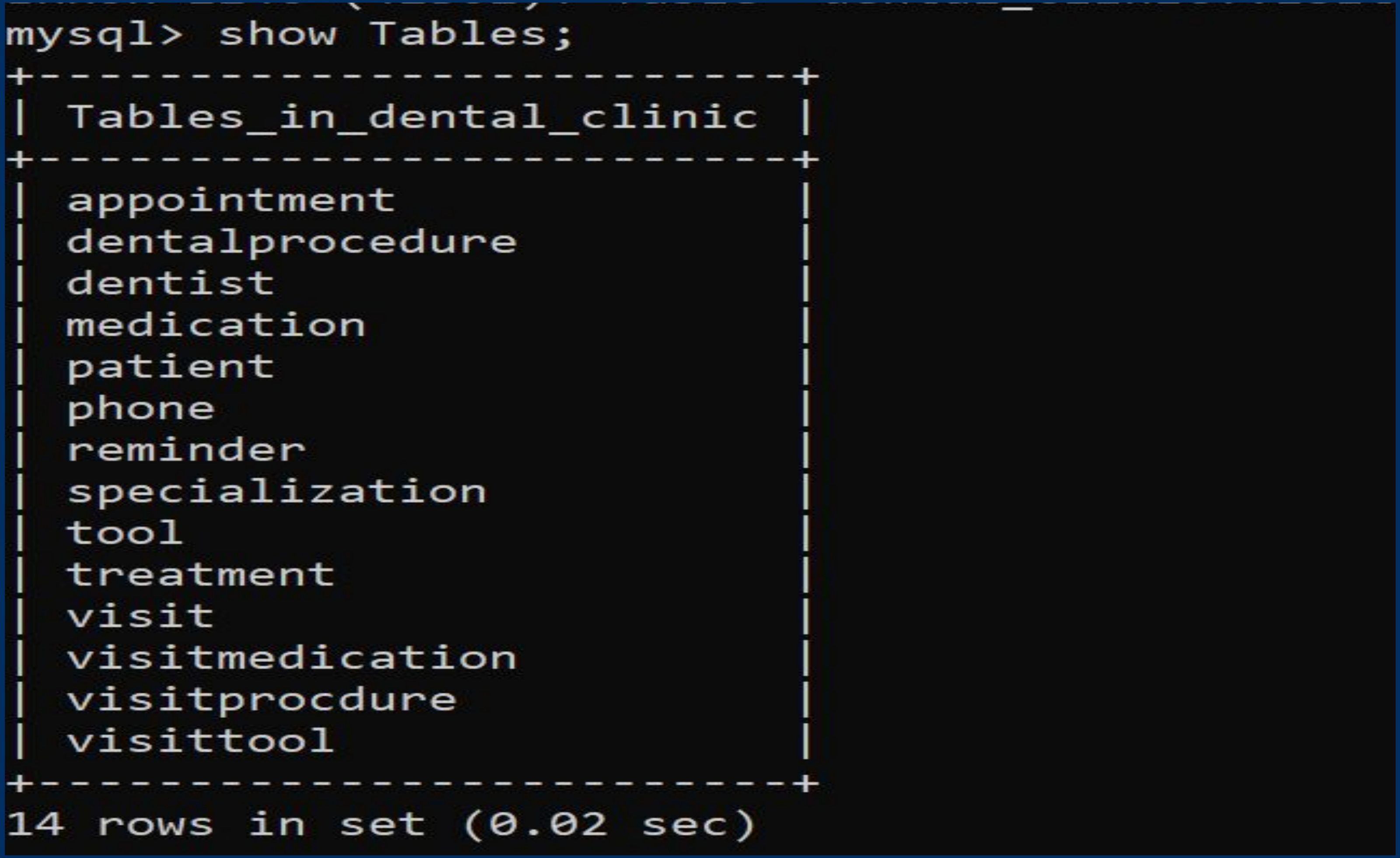
```
mysql> select * from phone;
+------+-------+
| p_id | ph_no |
+------+-------+
|    2 | 47573 |
|    1 | 27364 |
|    3 | 37468 |
+------+-------+
3 rows in set (0.00 sec)
```

SPECAILIZATION in Dentist:



```
mysql> select * from specialization;
+------+---------------------+
| d_id | specialization      |
+------+---------------------+
|    1 | Orthodontics        |
|    2 | Pediatric Dentistry |
|    3 | Endodontics         |
|    2 | Oral Surgery        |
+------+---------------------+
4 rows in set (0.00 sec)
```

ALL THE TABLES INCLUDED IN OUR DB:

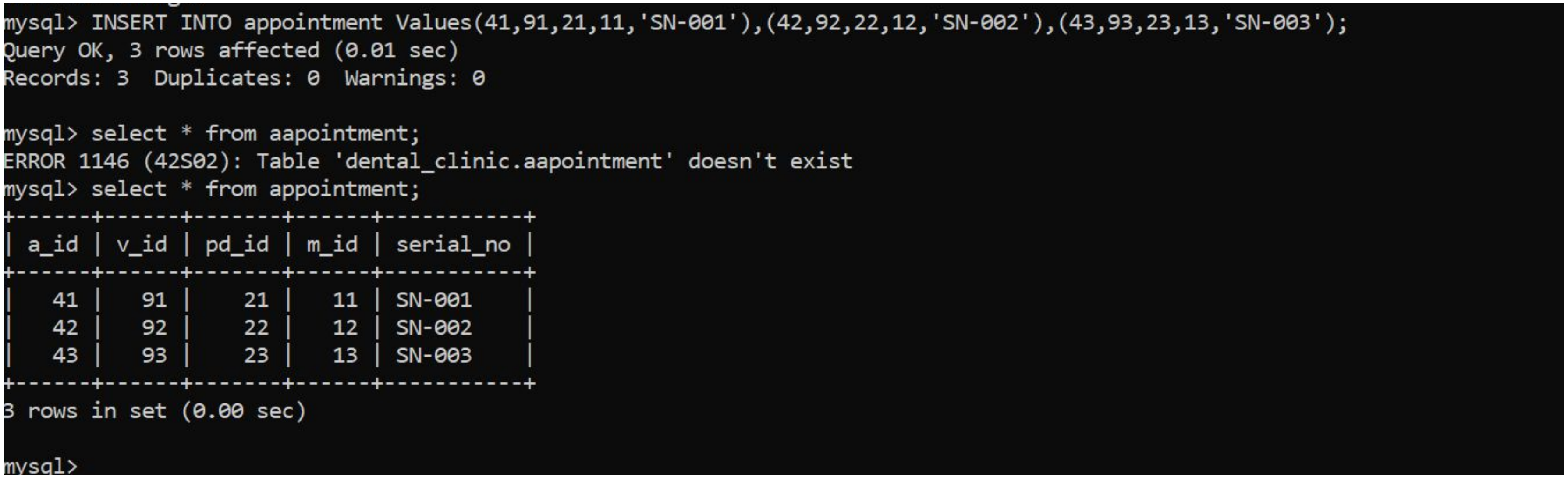


```
mysql> show Tables;
+--------------------------+
| Tables_in_dental_clinic |
+--------------------------+
| appointment              |
| dentalprocedure          |
| dentist                  |
| medication               |
| patient                  |
| phone                    |
| reminder                 |
| specialization           |
| tool                     |
| treatment                |
| visit                    |
| visitmedication          |
| visitprocdure            |
| visittool                |
+--------------------------+
14 rows in set (0.02 sec)
```

WITH THE HELP OF SHOW TABLES: YOU'LL AB ABLE TO SEE THE ALL THE TABLES WHICH IS PRESENT IN YOUR DATABASE .

INSERT COMMAND ON MYSQL;

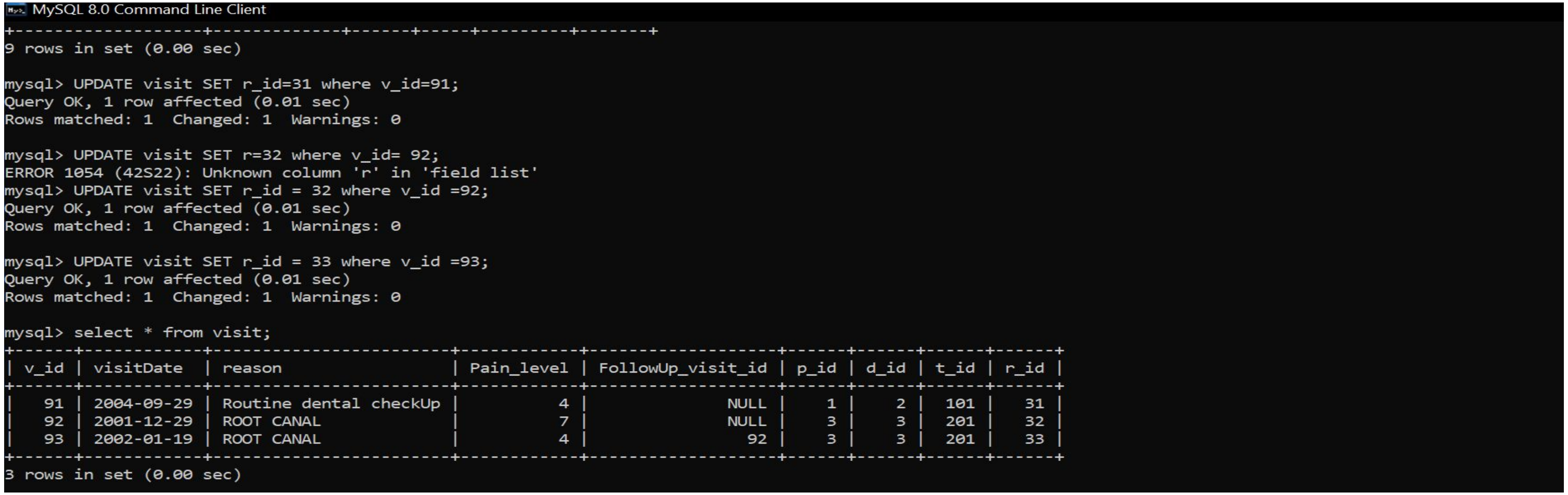
INSERT INTO table_name VALUES (…………………..);



```
mysql> INSERT INTO appointment Values(41,91,21,11,'SN-001'),(42,92,22,12,'SN-002'),(43,93,23,13,'SN-003');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from aapointment;
ERROR 1146 (42S02): Table 'dental_clinic.aapointment' doesn't exist
mysql> select * from appointment;
+------+------+-------+------+-----------+
| a_id | v_id | pd_id | m_id | serial_no |
+------+------+-------+------+-----------+
|   41 |   91 |    21 |   11 | SN-001    |
|   42 |   92 |    22 |   12 | SN-002    |
|   43 |   93 |    23 |   13 | SN-003    |
+------+------+-------+------+-----------+
3 rows in set (0.00 sec)

mysql>
```

UPDATE COMMAND:

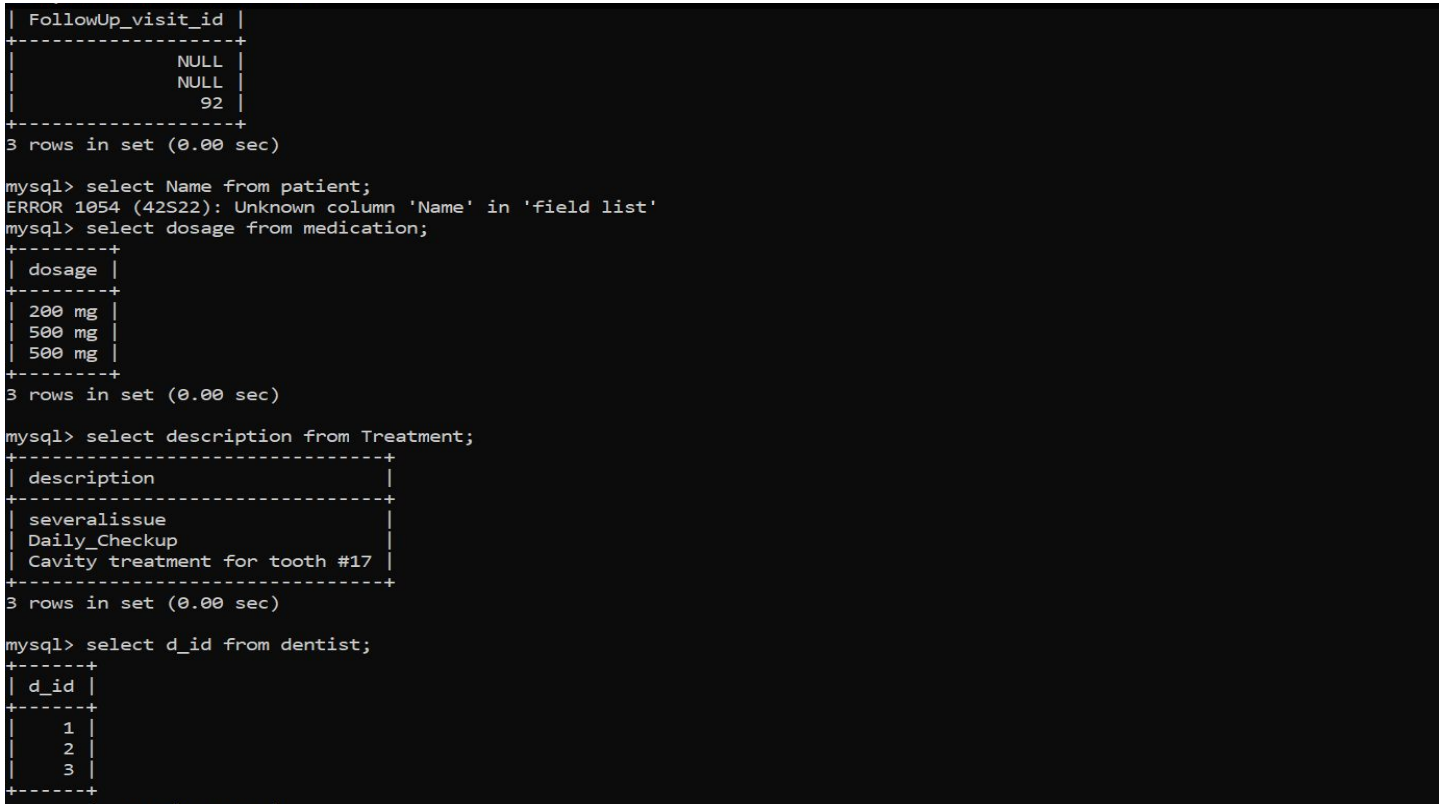
UPDATE table_name SET column_name=“ “ where column=“ “:



```
+-------------------+-------------+------+-----+---------+--------+
9 rows in set (0.00 sec)

mysql> UPDATE visit SET r_id=31 where v_id=91;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE visit SET r=32 where v_id= 92;
ERROR 1054 (42S22): Unknown column 'r' in 'field list'
mysql> UPDATE visit SET r_id = 32 where v_id =92;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

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

mysql> select * from visit;
+------+------------+------------------------+------------+-------------------+------+------+------+------+
| 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 |
|   93 | 2002-01-19 | ROOT CANAL             |          4 |                92 |    3 |    3 |  201 |   33 |
+------+------------+------------------------+------------+-------------------+------+------+------+------+
3 rows in set (0.00 sec)
```

TO SELECT A CERTAIN COUMN FROM THE TABLE:

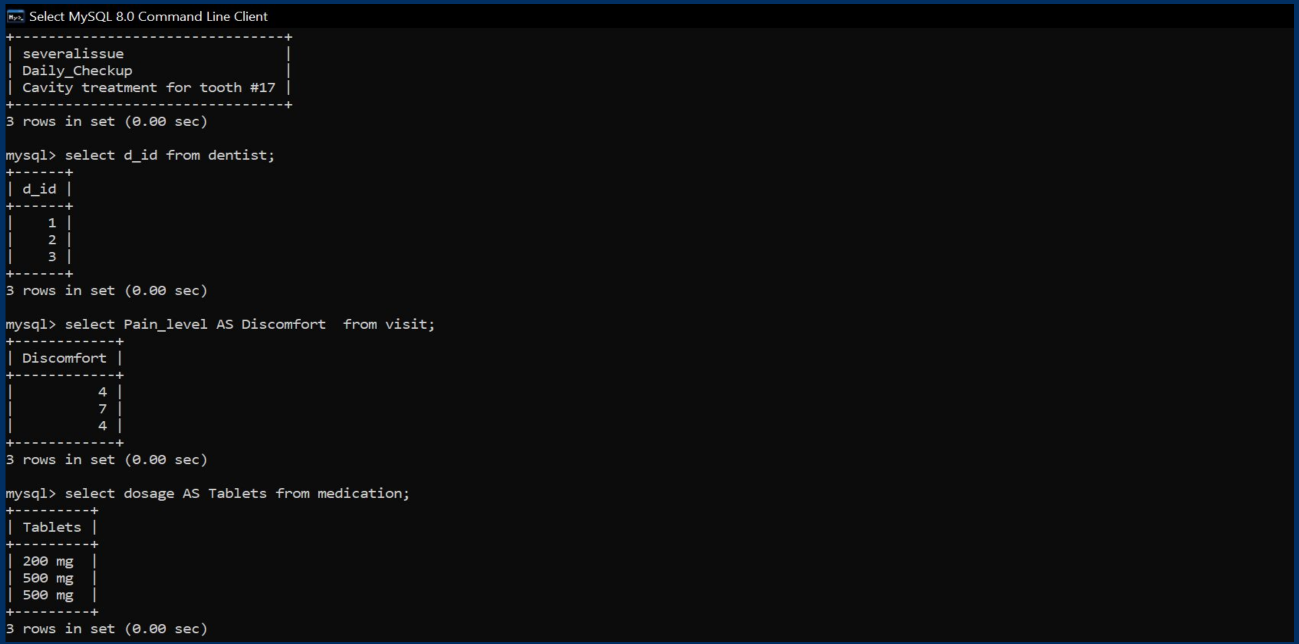
SELECT Column_name from table_name;



```
| FollowUp_visit_id |
+-------------------+
|              NULL |
|              NULL |
|                92 |
+-------------------+
3 rows in set (0.00 sec)

mysql> select Name from patient;
ERROR 1054 (42S22): Unknown column 'Name' in 'field list'
mysql> select dosage from medication;
+--------+
| dosage |
+--------+
| 200 mg |
| 500 mg |
| 500 mg |
+--------+
3 rows in set (0.00 sec)

mysql> select description from Treatment;
+--------------------------------+
| description                    |
+--------------------------------+
| severalissue                   |
| Daily_Checkup                  |
| Cavity treatment for tooth #17 |
+--------------------------------+
3 rows in set (0.00 sec)

mysql> select d_id from dentist;
+------+
| d_id |
+------+
|    1 |
|    2 |
|    3 |
+------+
```

SELECT NAME BY ALIAS :

SELECT original_name AS alias from table_name;



```
+--------------------------------+
| severalissue                   |
| Daily_Checkup                  |
| Cavity treatment for tooth #17 |
+--------------------------------+
3 rows in set (0.00 sec)

mysql> select d_id from dentist;
+------+
| d_id |
+------+
|    1 |
|    2 |
|    3 |
+------+
3 rows in set (0.00 sec)

mysql> select Pain_level AS Discomfort  from visit;
+------------+
| Discomfort |
+------------+
|          4 |
|          7 |
|          4 |
+------------+
3 rows in set (0.00 sec)

mysql> select dosage AS Tablets from medication;
+---------+
| Tablets |
+---------+
| 200 mg  |
| 500 mg  |
| 500 mg  |
+---------+
3 rows in set (0.00 sec)
```

CONCLUSION:

In conclusion, the dental clinic database effectively applies normalization principles to create a structured and efficient system for managing patient and clinic information. By organizing data into distinct entities and ensuring that multivalue attributes are handled appropriately, the database minimizes redundancy and enhances data integrity. This thoughtful design not only streamlines operations but also improves the overall quality of patient care, making it easier for the clinic to deliver exceptional service. Ultimately, the database serves as a vital tool in supporting the clinic's mission to provide effective and organized dental care.