key database objects you can create in mysql

Tables: The primary structure used to store data in rows and columns.
Views: Virtual tables created based on the result-set of an SQL statement. They do not contain data themselves but display data stored in tables.
Indexes: Used to retrieve data from the database more quickly than otherwise. They are typically created to improve the performance of frequently used queries.
Stored Procedures: A set of SQL statements that can be stored in the server and executed whenever you need to perform a particular task multiple times.
Functions: Similar to stored procedures, functions are routines that you can call to perform operations and return a value. They can be used in SQL statements anywhere an expression is allowed.
Triggers: These are automatic actions that are fired or executed when specific changes are made to a table, such as inserting, updating, or deleting records.
Events: Tasks that are scheduled to run at preset times or intervals, similar to cron jobs in Unix/Linux systems.
User-Defined Types (UDTs): Custom data types defined by users that can be used to handle complex data more flexibly.

MySQL ALTER TABLE – Add columns to a table

```
CREATE TABLE vehicles (
    vehicleId INT,
    year INT NOT NULL,
    make VARCHAR(100) NOT NULL,
    PRIMARY KEY(vehicleId)
);
```

MySQL ALTER TABLE – Add columns to a table

1) Add a column to a table

```
ALTER TABLE table_name
ADD
    new_column_name column_definition
    [FIRST | AFTER column_name]
```

In this syntax:

- `table_name` – specify the name of the table to which you want to add a new column or columns after the `ALTER TABLE` keywords.
- `new_column_name` – specify the name of the new column.
- `column_definition` – specify the datatype, maximum size, and column constraint of the new column
- `FIRST | AFTER column_name` specify the position of the new column in the table. You can add a column after an existing column (`ATER column_name`) or as the first column (`FIRST`). If you omit this clause, the column is appended at the end of the column list of the table.

ALTER TABLE vehicles
ADD model VARCHAR(100) NOT NULL;

DESCRIBE `vehicles;`

`2) Add multiple columns to a table`

```
ALTER TABLE table_name
    ADD new_column_name column_definition
    [FIRST | AFTER column_name],
    ADD new_column_name column_definition
    [FIRST | AFTER column_name],
    ...;
```

For example, this statement adds two columns `color` and `note` to the `vehicles` table:

ALTER TABLE vehicles
ADD color VARCHAR(50),
ADD note VARCHAR(255);

DESCRIBE vehicles;

MySQL ALTER TABLE – Modify columns

1) Modify a column

ALTER TABLE table_name
MODIFY column_name column_definition
[FIRST | AFTER column_name];

```
DESCRIBE vehicles;
```

```
Suppose that you want to change the note column a NOT NULL
column with a maximum of 100 characters.
```

```
ALTER TABLE vehicles
MODIFY note VARCHAR(100) NOT NULL;
```

```
DESCRIBE vehicles;
```

```
2) Modify multiple columns
```

```
ALTER TABLE table_name
    MODIFY column_name column_definition
    [ FIRST | AFTER column_name],
    MODIFY column_name column_definition
    [ FIRST | AFTER column_name],
    ...;
```

```
ALTER TABLE vehicles
MODIFY year SMALLINT NOT NULL,
MODIFY color VARCHAR(20) NULL AFTER make;
```

MySQL ALTER TABLE – Rename a column in a table

```
ALTER TABLE table_name
    CHANGE COLUMN original_name new_name column_definition
    [FIRST | AFTER column_name];
```

The following example uses the `ALTER TABLE CHANGE COLUMN` statement to rename the column `note` to `vehicleCondition`:

ALTER TABLE vehicles
CHANGE COLUMN note vehicleCondition VARCHAR(100) NOT NULL;

DESCRIBE vehicles;

MySQL ALTER TABLE – Drop a column

ALTER TABLE table_name
DROP COLUMN column_name;

This example shows how to remove the `vehicleCondition` column from the `vehicles` table:

ALTER TABLE vehicles
DROP COLUMN vehicleCondition;

MySQL ALTER TABLE – Rename table

ALTER TABLE table_name
RENAME TO new_table_name;

This example renames the `vehicles` table to `cars`:

ALTER TABLE vehicles
RENAME TO cars;

The MySQL UPDATE Statement

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

MySQL | Grant / Revoke Privilege

Privileges: The privileges that can be granted to the users are listed below along with the description:

Privilege	Description
SELECT	*select statement on tables*
INSERT	*insert statement on the table*
DELETE	*delete statement on the table*
INDEX	*Create an index on an existing table*
CREATE	*Create table statements*
ALTER	*Ability to perform ALTER TABLE to change the table definition*
DROP	*Drop table statements*
ALL	*Grant all permissions except GRANT OPTION*
UPDATE	*Update statements on the table*
GRANT	*Allows to grant the privilege that*

Grant Privileges on Table