```
create table mytab_3(
    -> id int primary key,
    -> name varchar(20));
Query OK, 0 rows affected (0.03 sec)

mysql> create table mychild_3(
    -> cid int primary key,
    -> data int,
    -> constraint fk_id foreign key(data) references mytab_3(id)
    -> on update cascade
    -> on delete set null);
Query OK, 0 rows affected (0.03 sec)
select * from course;
+-----+-------+------+
| cid | cname | size |
+-----+-------+------+
|  10 | java  |  240 |
|  11 | cpp   |  400 |
+-----+-------+------+
2 rows in set (0.00 sec)

mysql> select * from faculty;
+-----+------------+------------+
| fid | fname      | experience |
+-----+------------+------------+
| 100 | Bakul      |         15 |
| 101 | Dhanashree |          7 |
+-----+------------+------------+
2 rows in set (0.00 sec)

mysql> insert into course_faculty values(10,100)
```

```
    -> ;
Query OK, 1 row affected (0.01 sec)


mysql> insert into course_faculty values(10,101)
    -> ;
Query OK, 1 row affected (0.00 sec)


mysql> insert into course_faculty values(11,101)
    -> ;
Query OK, 1 row affected (0.00 sec)


mysql> insert into course_faculty values(10,101)
    -> ;
ERROR 1062 (23000): Duplicate entry '10-101' for key 'course_faculty.PRIMARY'
mysql> select * from course_faculty;
+-----+-----+
| cid | fid |
+-----+-----+
|  10 | 100 |
|  10 | 101 |
|  11 | 101 |
+-----+-----+
3 rows in set (0.00 sec)


mysql> update course
    -> set cid=25
    -> where cid=10;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0


mysql> select * from course;
```

```
+-----+-------+------+
| cid | cname | size |
+-----+-------+------+
|  11 | cpp   |  400 |
|  25 | java  |  240 |
+-----+-------+------+
2 rows in set (0.00 sec)

mysql> select * from course_faculty;
+-----+-----+
| cid | fid |
+-----+-----+
|  25 | 100 |
|  11 | 101 |
|  25 | 101 |
+-----+-----+
3 rows in set (0.00 sec)

mysql> delete from faculty
    -> where fid=100;
Query OK, 1 row affected (0.01 sec)

mysql> select * from course_faculty;
+-----+-----+
| cid | fid |
+-----+-----+
|  11 | 101 |
|  25 | 101 |
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> create table mytab_3(
    -> id int primary key,
    -> name varchar(20));
ERROR 1050 (42S01): Table 'mytab_3' already exists
mysql> drop table mytab_3;
Query OK, 0 rows affected (0.02 sec)

mysql> create table mytab_3(
    -> id int primary key,
    -> name varchar(20));
Query OK, 0 rows affected (0.03 sec)

mysql> create table mychild_3(
    -> cid int primary key,
    -> data int,
    -> constraint fk_id foreign key(data) references mytab_3(id)
    -> on update cascade
    -> on delete set null);
Query OK, 0 rows affected (0.03 sec)

mysql> insert into mytab_3 values(10,'xxx');
Query OK, 1 row affected (0.01 sec)

mysql> insert into mytab_3 values(11,'yyyy');
Query OK, 1 row affected (0.00 sec)

mysql> insert into mychild_3 values(1,10);
Query OK, 1 row affected (0.01 sec)

mysql> insert into mychild_3 values(2,10);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into mychild_3 values(3,11);
Query OK, 1 row affected (0.01 sec)

mysql> insert into mychild_3 values(4,11);
Query OK, 1 row affected (0.01 sec)

mysql> select * from mytab_3;
+----+------+
| id | name |
+----+------+
| 10 | xxx  |
| 11 | yyyy |
+----+------+
2 rows in set (0.00 sec)

mysql> select * from mychild_3;
+-----+------+
| cid | data |
+-----+------+
|   1 |   10 |
|   2 |   10 |
|   3 |   11 |
|   4 |   11 |
+-----+------+
4 rows in set (0.00 sec)

mysql> update mytab_3
    -> set id=1000 where id=10;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from mychild_3;
+-----+------+
| cid | data |
+-----+------+
|   3 |   11 |
|   4 |   11 |
|   1 | 1000 |
|   2 | 1000 |
+-----+------+
4 rows in set (0.00 sec)

mysql> update mytab_3
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to
your MySQL server version for the right syntax to use near '' at line 1
mysql> delete from mytab_3
    -> where id=1000;
Query OK, 1 row affected (0.01 sec)

mysql> select * from mytab_3;
+----+------+
| id | name |
+----+------+
| 11 | yyyy |
+----+------+
1 row in set (0.00 sec)

mysql> select * from mychild_3;
+-----+------+
| cid | data |
+-----+------+
|   1 | NULL |
|   2 | NULL |
|   3 |   11 |
|   4 |   11 |
+-----+------+
4 rows in set (0.00 sec)
```

To insert data in the table

```
Insert into dept values(100,’HR’,’Pune’);
Insert into dept(deptno,dname) values(101,’admin’)
INSERT INTO Worker
        (WORKER_ID, FIRST_NAME, LAST_NAME, SALARY, JOINING_DATE, DEPARTMENT) VALUES
                (001, 'Monika', 'Arora', 100000, '14-02-20 09.00.00', 'HR'),
                (002, 'Niharika', 'Verma', 80000, '14-06-11 09.00.00', 'Admin'),
                (003, 'Vishal', 'Singhal', 300000, '14-02-20 09.00.00', 'HR'),
                (004, 'Amitabh', 'Singh', 500000, '14-02-20 09.00.00', 'Admin'),
                (005, 'Vivek', 'Bhati', 500000, '14-06-11 09.00.00', 'Admin'),
                (006, 'Vipul', 'Diwan', 200000, '14-06-11 09.00.00', 'Account'),
                (007, 'Satish', 'Kumar', 75000, '14-01-20 09.00.00', 'Account'),
                (008, 'Geetika', 'Chauhan', 90000, '14-04-11 09.00.00', 'Admin');
```

Nested query

- All the columns in the o/p will be added in course_backup table
- Constraints will not get copied

```
Create table course_backup as
(select * from course);
```

Delete the data

1. To delete all rows from the table
   Delete from course;
2. Delete all courses with cid<100;

Delete from course
Where cid<100;

To update the record

1. To update sal to 3000 and deptno to 30 for all employees with commision is not null
   update emp
   set sal=sal+3000,deptno=30
   where comm is not null;

using alter table

This statement can be used to add , delete, modify or rename a column

or you can rename table

or you can add constraint or drop constraint.

1. Add column

ALTER TABLE table_name

ADD new_column_name column_definition

[FIRST | AFTER column_name],

ADD new_column_name column_definition

[FIRST | AFTER column_name],

2. Drop column
   ALTER TABLE table_name
   Drop column <column name>
3. Modify column

ALTER TABLE table_name

modify <existing column> <new column name> datatype

4. Rename column

ALTER TABLE table_name

Change column <old column> <new column name> datatype

5. Add constraint
   Alter table <table name>
   Add constraint < name> primary key(id,name,desg);

   Alter table <table name>
   Add constraint < name> foreign key(field name) references <table name>(column name);

6. Drop constraint
   -----drop primary key
   Alter table <table name>
   Drop primary key
   ------drop foreign key
   Alter table <table name>
   Drop constraint <constraint name>
7. To rename table
   Alter table <table name>
   Rename to <new table name>

   To delete table data and structure
   Drop table <tablename>

   To find create table quey
   Show create table student;

   To delete all rows from the table but empty table structure should be there

   Truncate table <table name>

Differences between truncate and delete

Truncate	delete
It is DDL(data definition language) statement	It is DML(data manipulation language)
Rollback is not possible once data is deleted	Rollback is possible
Cannot use where clause	We may use where clause

Set autocommit=0; ---- set autocommit off
TCL (transaction control statement)
Commit---- it make the changes done by all DML statements permanent
Rollback---- it allows you to undo the changes till previous commit
Savepoint--- adds marker in between changes

All DDL operations are auto committed

Table ---- 10 rows-commited
Insert 1
Insert 1
commit
Insert 1
Delete 1
rollback

Table ---- 10 rows-commited
Insert 1
Insert 1
Alter table -------
Insert 1
Insert 1
Insert 1
Savepoint A
Delete 1
Insert 1
Insert 1
Savepoint B
Insert 1
Insert 1
Rollback to B

Case statement

If comm is null   ‘need improvement’
If <500 “ok”
If >=500 and < 1000   ‘good
‘Excellent’

```
select empno,ename,comm, case when comm is null then 'need improvement'
    -> when comm=0 then 'need improvement'
    -> when comm<500 then ' ok'
    -> when comm>=500 and comm <1000 then 'good'
    -> else 'excellent' end  remark
    -> from emp;
```

Nested query

1. Display all employees with sal > smith’s salary

```
Select sal
From emp
Where ename=’SMITH’;


Select *
From emp
Where sal > (Select sal
From emp
Where ename=’SMITH’;
)
```

2. Find all employees who are working in the same department of either SMITH or ALLEN

```
Select *
From emp
Where deptno in (Select deptno
                  From emp
                 Where ename in (‘SMITH’,’ALLEN’);

               )
```

3. Find all employees with sal > smiths sal and < Millers sal

```
Select *
From emp
Where sal between (select sal from emp where ename=’SMITH’) and
   (select sal from emp where ename=’MILLER’);
```