Database - Employee

```
tables - employees
         - salaries
         - titles
         - dept_manager
         - departments
         - dept_emp
```

To create Employee database

```
      - create database Employee;
```

To use a database

```
      -use Employee;
```

To create table employees

```
CREATE TABLE employees (
    emp_no  INT  NOT NULL AUTO_INCREMENT,  -- UNSIGNED AUTO_INCREMENT??
    birth_date  DATE             NOT NULL,
    first_name  VARCHAR(14)      NOT NULL,
    last_name   VARCHAR(16)      NOT NULL,
    gender      ENUM ('M','F')   NOT NULL,  -- Enumeration of either 'M'
or 'F'
    hire_date   DATE             NOT NULL,
    PRIMARY KEY (emp_no)                    -- Index built automatically
on primary-key column
      );
```

To create salaries table

```
    CREATE TABLE salaries (
    emp_no      INT    NOT NULL,
    salary      INT    NOT NULL,
    from_date   DATE   NOT NULL,
    to_date     DATE   NOT NULL,
    KEY         (emp_no),
    FOREIGN KEY (emp_no) REFERENCES employees (emp_no),
    PRIMARY KEY (emp_no, from_date)
      );
```

To create departments table

```
 CREATE TABLE departments (
    dept_no     CHAR(4)          NOT NULL,  -- in the form of 'dxxx'
    dept_name   VARCHAR(40)      NOT NULL,
    PRIMARY KEY (dept_no),                  -- Index built automatically
    UNIQUE  KEY (dept_name)                 -- Build INDEX on this
unique-value column
      );
```

To create dept_emp table

```
      -CREATE TABLE dept_emp (
    emp_no      INT         NOT NULL,
    dept_no     CHAR(4)     NOT NULL,
    from_date   DATE        NOT NULL,
    to_date     DATE        NOT NULL,
    KEY         (emp_no),   -- Build INDEX on this non-unique-value
column
    KEY         (dept_no),  -- Build INDEX on this non-unique-value
column
    FOREIGN KEY (emp_no) REFERENCES employees (emp_no) ,
    FOREIGN KEY (dept_no) REFERENCES departments (dept_no) ,
    PRIMARY KEY (emp_no, dept_no)
      );
```

To create dept_manager table

```
      - CREATE TABLE dept_manager (
   dept_no      CHAR(4)  NOT NULL,
   emp_no       INT      NOT NULL,
   from_date    DATE     NOT NULL,
   to_date      DATE     NOT NULL,
   KEY         (emp_no),
   KEY         (dept_no),
   FOREIGN KEY (emp_no)  REFERENCES employees (emp_no) ,
   FOREIGN KEY (dept_no) REFERENCES departments (dept_no) ,
   PRIMARY KEY (emp_no, dept_no)
      );
```

To create titles table

```
      -CREATE TABLE titles (
    emp_no      INT          NOT NULL,
    title       VARCHAR(50)  NOT NULL,
    from_date   DATE         NOT NULL,
    to_date     DATE,
    KEY         (emp_no),
    FOREIGN KEY (emp_no) REFERENCES employees (emp_no) ,
    PRIMARY KEY (emp_no, title, from_date)
      );
```

To insert values to tables

```
      -insert into employees(emp_no, birth_date, first_name, last_name,
gender, hire_date) values (0001, 19-03-18, 'naveen', 'karthik', 'M',
'13-12-23');
      -insert into departments values(1, 'Civil department');
      -insert into dept_emp vales(1,1, '13-12-23', '13-12-25');
      -insert into dept_manager values(1, 1, '13-12-23', '13-12-25');
      -insert into titles values(1, Manager,'13-12-23', '13-12-25');
      -insert into salaries values(1, 50000, '13-12-23', '13-12-25');
```

To update the tables

```
      -update employees set first_name='kavin' where emp_no=1;
```

To delete a row in the table

```
      -delete from employees where emp_no=1;
```

```
To delete all records in a table
      -delete from employees;

To delete a table
      -drop table employees;

To select all employees from the table employees
      - select * from employees;

To select particular columns from the table employees
      - select first_name,hire_date,gender from employees;

To select unique or distinct values from the table
      - select distinct first_name from employees;
      - select distinct title from titles;
      - select distinct dept_name from departments;

To select employee with some conditions
      -select * from employees where gender = 'M';

Order employees by their hire_date
      -select * from employees order by hire_date;
      -select * from salaries order by salary desc; -- it hely to sort
salaries table in descending order.

To add multiple conditions using add keyword
      -select * from employees where first_name='fname' and
last_name='lname';

To sort the table with one or more conditions using or keyword
      - select * from employees where gender ='M' or hire_date='date';

To sort the table wit not keywork
      - select * from salaries where  not salary=10000;

To sort limited records from the table
      -select * from employees limit 5;

Using max() and min() funtions
      - select max(salary) from salaries;
      - select min(salary) from salaries;
      - select min(salary) as lower_salary from salaries;

Using cont()
      - select count(*) from employees;  --used to count th no.of
employees
      - select count(*) from employees where gender = 'M';

Using sum()
      - select sum(salary) from salaries; --used to get sum of salary
      - select sum(salary) from salaries where from_date='21-06-22';

Using avg()
      - select avg(salary) from salaries;
      - select avg(salary) from employees where from_date='21-06-22';

Using Like operators
      -SELECT * FROM employees WHERE first_name LIKE 'a%';
      -SELECT * FROM employees WHERE first_name LIKE '%a%';
      -SELECT * FROM employees WHERE first_name LIKE '%a';
      -SELECT * FROM employees WHERE first_name LIKE 'a__%';
      -SELECT * FROM employees WHERE first_name LIKE '%__a;

Using In operator
      - select * from salaries where salary in (10000,20000,30000);
      - select * from salaries where salary not in (10000,20000,30000);

Using between operator
      - select * from salaries where salary between 10000 and 30000;
      - select * from salaries where salary not between 10000 and
30000;

Using as keyword
      - select first_name as name from employees;

Using Inner joins
      - select * from employees inner join salaries on employees.emp_no
= salaries.emp_no;
      - select em.first_name, sa.salary from employees as em inner join
salaries as sa on em.emp_no = sa.emp_no;

Using Left joins
      - select * from employees left join salaries on employees.emp_no
= salaries.emp_no;
      - select em.first_name, sa.salary from employees as em left join
salaries as sa on em.emp_no = sa.emp_no;

Using right joins
      - select * from employees right join salaries on employees.emp_no
= salaries.emp_no;
      - select em.first_name, sa.salary from employees as em right join
salaries as sa on em.emp_no = sa.emp_no;

Using cross join
      - select * from employees cross join salaries on employees.emp_no
= salaries.emp_no;
      - select em.first_name, sa.salary from employees as em cross join
salaries as sa on em.emp_no = sa.emp_no;

Using self join
```

```
      - select * from employees, salaries where employees.emp_no =
salaries.emp_no;
      - select em.first_name, sa.salary from employees as em, salaries
as sa where em.emp_no = sa.emp_no;

Using group by keyword
      - select count(emp_no), title from titles group by title;

Using keyword having;
      - select count(emp_no),title from titles group by title having
title='manager';

Using exists keyword
      - select first_name,salary from employees, salaries where
exists(select salary from salaries where salaries.emp_no=
employees.emp_no and salary > 200000);

Using any and all keyword
      - select first_name from employees where emp_no = any( select
emp_no from titles where title = 'manager');
      - select first_name from employees where emp_no = all( select
emp_no from titles where title = 'manager');

Using insert into keyword
      - insert into employees select * from employees_old; --
employees_old is another database where old employee datas are stored.

Using case statements
      - select emp_no, gender case when gender = 'M' then 'Work from
home' when gender='F' then 'work from office' else 'there is no
employee' end as type_of_work from employees;

To alter table
      - alter table employees add address varchar(255);
      - alter table employees drop adress;
      - alter table employees modify emp_no varchar(10);
      - alter table employees change first_name f_name varchar(20);

Using check and default
      - create table employee_details(
      emp_no int primary key,
      mobile_num int(15) not null default 0000000000,
      age int,
      check( age > 18)
      );

Using date, datetime
      - select * from employees where hire_date ='2023-11-16';
      - select * from employees where hire_date >'2023-11-16';
      - select * from employees where hire_date <'2023-11-16';
      - select * from employees where hire_date >'2023-11-16' and
hire_date < '2023-12-10';
      - SELECT * FROM employees where  hire_date between '1997-01-00'
and '1997-01-31';
      - SELECT * FROM employees where month(hire_date) = '02';
      - select * from employees where month(hire_date) = '02' and
year(hire_date) = '2023';
      - select * from employees where year(hire_date) = '2020';
      - select * from employees WHERE hire_date BETWEEN CURDATE() -
INTERVAL 1 DAY AND CURDATE();

      - select * from employees where time(hire_date) = '20:00:00';
      - select * from employees where time(hire_date) > '20:00:00';
      - select * from employees where time(hire_date) < '20:00:00';
      - select * from employees where time(hire_date) >'18:00:00' and
time(hire_date) < '20:00:00';
      - SELECT * FROM employees where  time(hire_date) between
'18:00:00' and '18:00:00';


To drop a column in a table
      - alter table table_name drop column column_name;


difference between delete and truncate?

using truncate
      - truncate table_name;

Nested query or Subquery

      - select * from employees where hire_date in (select from_date
from dept_emp);
```