**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);