--creating and inserting data into tables

create table students

(

student\_no integer,

student\_name varchar(20),

age integer

);

insert into students values (1, 'Michael', 19);

insert into students values (2, 'Doug', 18);

insert into students values (3, 'Samantha', 21);

insert into students values (4, 'Pete', 20);

insert into students values (5, 'Ralph', 19);

insert into students values (6, 'Arnold', 22);

insert into students values (7, 'Michael', 19);

insert into students values (8, 'Jack', 19);

insert into students values (9, 'Rand', 17);

insert into students values (10, 'Sylvia', 20);

create table courses

(

course\_no varchar(5),

course\_title varchar(20),

credits integer

);

insert into courses values ('CS110', 'Pre Calculus', 4);

insert into courses values ('CS180', 'Physics', 4);

insert into courses values ('CS107', 'Intro to Psychology', 3);

insert into courses values ('CS210', 'Art History', 3);

insert into courses values ('CS220', 'US History', 3);

create table student\_enrollment

(

student\_no integer,

course\_no varchar(5)

);

insert into student\_enrollment values (1, 'CS110');

insert into student\_enrollment values (1, 'CS180');

insert into student\_enrollment values (1, 'CS210');

insert into student\_enrollment values (2, 'CS107');

insert into student\_enrollment values (2, 'CS220');

insert into student\_enrollment values (3, 'CS110');

insert into student\_enrollment values (3, 'CS180');

insert into student\_enrollment values (4, 'CS220');

insert into student\_enrollment values (5, 'CS110');

insert into student\_enrollment values (5, 'CS180');

insert into student\_enrollment values (5, 'CS210');

insert into student\_enrollment values (5, 'CS220');

insert into student\_enrollment values (6, 'CS110');

insert into student\_enrollment values (7, 'CS110');

insert into student\_enrollment values (7, 'CS210');

create table professors

(

last\_name varchar(20),

department varchar(12),

salary integer,

hire\_date date

);

insert into professors values ('Chong', 'Science', 88000, '2006-04-18');

insert into professors values ('Brown', 'Math', 97000, '2002-08-22');

insert into professors values ('Jones', 'History', 67000, '2009-11-17');

insert into professors values ('Wilson', 'Astronomy', 110000, '2005-01-15');

insert into professors values ('Miller', 'Agriculture', 82000, '2008-05-08');

insert into professors values ('Williams', 'Law', 105000, '2001-06-05');

create table teach

(

last\_name varchar(20),

course\_no varchar(5)

);

insert into teach values ('Chong', 'CS180');

insert into teach values ('Brown', 'CS110');

insert into teach values ('Brown', 'CS180');

insert into teach values ('Jones', 'CS210');

insert into teach values ('Jones', 'CS220');

insert into teach values ('Wilson', 'CS110');

insert into teach values ('Wilson', 'CS180');

insert into teach values ('Williams', 'CS107');

-- wild card

SELECT \* FROM employees;

-- specific columns

SELECT employee\_id, first\_name, department

FROM employees;

-- use WHERE to filter query (remember single quotes)

-- sql case-insensitive, but data is sensitive

SELECT \*

FROM employees

WHERE department = 'Sports';

--use 'like' keyword for a rough match with part of the desired match

SELECT \*

FROM employees

WHERE department like 'F%ture%'; -- returns furniture

--filter with numbers

SELECT \*

FROM employees

WHERE salary > 100000;

-- WHERE with AND for further filtering

SELECT \*

FROM employees

WHERE department = 'Clothing'

AND salary > 100000

AND region\_id = 2;

-- use OR for multiple conditions

SELECT \*

FROM employees

WHERE department = 'Clothing'

AND salary > 100000

OR region\_id = 2;

-- combining AND and OR.. grouping symbols ensure prioity ooo

SELECT \*

FROM employees

WHERE salary < 40000

AND (department = 'Clothing'

OR region\_id = 2);

-- NOT Operator

SELECT \*

FROM employees

WHERE NOT department = 'Sports';

-- bang operator

SELECT \*

FROM employees

WHERE department != 'Furniture';

-- or use angle brackets to filter for everything except...

SELECt \*

FROM employees

WHERE department <> 'Clothing';

-- null values

SELECT \*

FROM employees

WHERE email IS NULL;

SELECT \*

FROM employees

WHERE NOT email IS NULL;

--same as above

SELECT \*

FROM employees

WHERE email IS NOT NULL;

-- using IN

SELECT \*

FROM employees

WHERE department IN ('Sports', 'Furniture', 'Clothing')

AND email IS NOT NULL;

-- filter by range values... BETWEEN is inclusive

SELECT \*

FROM employees

WHERE salary BETWEEN 80000 AND 100000;

-- practice queries

SELECT first\_name, email

FROM employees

WHERE gender = 'F'

AND department = 'Tools'

AND salary > 110000;

SELECT first\_name, hire\_date

FROM employees

WHERE salary > 165000

OR (department = 'Sports' AND gender = 'M');

SELECT first\_name, hire\_date

FROM employees

WHERE hire\_date BETWEEN '2002-01-01' AND '2004-01-01';

SELECT \*

FROM employees

WHERE (gender = 'M' AND department = 'Automotive' AND salary BETWEEN 40000 AND 100000)

OR (gender = 'F' AND department = 'Toys');

-- ORDER BY allows for sorting of returned table -numerical ascending by default

SELECT \*

FROM employees

ORDER BY employee\_id;

--sort descending

SELECT \*

FROM employees

ORDER BY employee\_id desc;

--alphabetical ascending by default for String values

SELECT \*

FROM employees

ORDER BY department;

-- get unique values using DISTINCT

SELECT DISTINCT department

FROM employees

ORDER BY 1; --orders by the departments default

--use limit to get a sample of the responses

SELECT DISTINCT department

FROM employees

ORDER BY 1

LIMIT 10;

--same result using fetch (changed to desc)

SELECT DISTINCT department

FROM employees

ORDER BY 1 desc

FETCH FIRST 10 ROWS ONLY;

--rename resultant column

SELECT DISTINCT department as sorted\_departments

FROM employees

ORDER BY 1 desc

FETCH FIRST 10 ROWS ONLY;

--rename columns for export quality

SELECT first\_name as "First Name", salary as "Yearly Salary"

FROM employees

ORDER BY salary DESC

FETCH FIRST 10 ROWS ONLY;

--query practice 2

SELECT \*

FROM students

WHERE age BETWEEN 18 AND 20;

SELECT \*

FROM students

WHERE student\_name like '%ch%'

OR student\_name LIKE '%nd';

SELECT \*

FROM students

WHERE(student\_name LIKE '%ae%' OR student\_name LIKE '%ph%')

AND age != 19;

SELECT student\_name

FROM students

ORDER BY age DESC;

SELECT student\_name, age

FROM students

ORDER BY age DESC

LIMIT 4;

SELECT \*

FROM students

WHERE (age < 20 AND (student\_no BETWEEN 3 AND 5 OR student\_no = 7))

OR age > 20 and student\_no > 3;