--Reorder columns

SELECT first\_name, last\_name, \* FROM employees;

--selecting from multiple tables

SELECT first\_name, last\_name, \*

FROM employees, departments;

--but beware selecting a column that exists in both tables

--produces error (ambiguous)

-- SELECT department

-- FROM employees, department;

--instead use dot notation

SELECT employees.department

FROM employees, departments;

SELECT departments.department

FROM employees, departments;

--aliases

SELECT d.department

FROM employees, departments d;

--Subqueries, a query within a query

--Subquery in WHERE clause

--returns the employees not assigned to a specific department in the departments table

SELECT \* FROM employees

WHERE department NOT IN(SELECT department from departments);

--camping, security, plumbing, maintenance

--Subquery in FROM clause...requires an alias

SELECT \*

FROM(SELECT \* FROM employees WHERE salary > 150000) as a;

--use subquery alias in select...but column names in subquery must match column names in select statement

SELECT a.first\_name, a.salary

FROM (SELECT \* FROM employees WHERE salary > 150000) as a;

SELECT employee\_name, annual\_salary

FROM (SELECT first\_name employee\_name, salary annual\_salary

FROM employees WHERE salary > 150000) a;

--Subqueries in the SELECT clause

SELECT first\_name, last\_name, salary, (SELECT first\_name FROM employees LIMIT 1)

FROM employees;

SELECT \* FROM regions

--Suquery with 'IN'

SELECT first\_name, last\_name, department

FROM employees

WHERE department IN (SELECT department

FROM departments WHERE division = 'Electronics');

SELECT first\_name, last\_name, salary, region\_id

FROM employees

WHERE salary > 130000

AND region\_id IN (SELECT region\_id FROM regions

WHERE region\_id>=4);

--OR...

SELECT \* FROM employees

WHERE salary > 130000

AND region\_id IN (SELECT region\_id FROM regions

WHERE country IN ('Asia', 'Canada'));

SELECT first\_name, department,

(SELECT MAX(salary)from employees) -salary as difference

FROM employees

WHERE region\_id IN (SELECT region\_id FROM regions

WHERE country IN ('Asia', 'Canada'));

SELECT \* FROM employees

WHERE region\_id IN (SELECT region\_id FROM regions WHERE country = 'United States');

-- Subqueries using ALL/ANY

SELECT \* FROM employees

WHERE region\_id > ALL (SELECT region\_id FROM regions WHERE country = 'United States');

SELECT \* FROM employees

WHERE department IN (SELECT department from departments

WHERE division = 'Kids')

AND hire\_date > ALL (SELECT hire\_date from employees

WHERE department = 'Maintenance') ;

--ANother Solution

SELECT \* from employees

WHERE department = ANY (SELECT department from departments

WHERE division = 'Kids')

AND hire\_date > ALL (SELECT hire\_date from employees

WHERE department = 'Maintenance') ;

--get most frequent salary (highest for tie)

SELECT salary, count(\*)

FROM employees

GROUP BY salary

ORDER BY count DESC, salary DESC LIMIT 1;

--getting only salary, use subquery

SELECT salary FROM

(SELECT salary, count(\*)

FROM employees

GROUP BY salary

ORDER BY count DESC, salary DESC LIMIT 1) a;

--Another Solution

SELECT salary

FROM employees

GROUP BY salary

HAVING COUNT(\*) >= ALL (SELECT count(\*) FROM employees

GROUP BY salary)

ORDER BY salary DESC

LIMIT 1;

--PRACTICE PROBLEMS

DROP TABLE dupes;

CREATE TABLE dupes(id integer, name varchar(10));

INSERT INTO dupes VALUES(1, 'FRANK');

INSERT INTO dupes VALUES(2, 'FRANK');

INSERT INTO dupes VALUES(3, 'ROBERT');

INSERT INTO dupes VALUES(4, 'ROBERT');

INSERT INTO dupes VALUES(5, 'SAM');

INSERT INTO dupes VALUES(6, 'FRANK');

INSERT INTO dupes VALUES(7, 'PETER');

--Distinct names and IDs

SELECT MIN(id), name

FROM dupes

GROUP BY name;

--USE DELETE to keep only these unique records

DELETE FROM dupes

WHERE id NOT IN

(SELECT MIN(id)

FROM dupes

GROUP BY name)

SELECT \* FROM dupes;

--Back to employees; mean salary removing potential outliers

SELECT ROUND(AVG(salary))

FROM employees

WHERE salary NOT IN((SELECT MIN(salary) FROM employees),

(SELECT MAX(salary) FROM employees

))

--ASSIGNMENT

--MINE

SELECT student\_name

FROM students

WHERE student\_no IN (SELECT student\_no from student\_enrollment

WHERE course\_no = 'CS180' OR course\_no = 'CS220');

--SOLUTION (SAME RESULT)

SELECT student\_name

FROM students WHERE student\_no

IN (SELECT student\_no

FROM student\_enrollment

WHERE course\_no

IN ( SELECT course\_no

FROM courses

WHERE course\_title

IN ('Physics', 'US History')));

--MINE

SELECT \*

FROM students

WHERE student\_no IN (SELECT student\_no

FROM student\_enrollment

GROUP BY student\_no

ORDER BY count(student\_no) DESC

LIMIT 1)

--SOLUTION(same result)

SELECT \*

FROM students

WHERE student\_no IN (SELECT student\_no FROM (SELECT student\_no, COUNT(course\_no) course\_cnt FROM STUDENT\_ENROLLMENT GROUP BY student\_no ORDER BY course\_cnt desc LIMIT 1 )a )

--MINE

SELECT \*

FROM students

WHERE age >= ALL(SELECT MAX(age) from students);

--Solution

SELECT \*

FROM students

WHERE age = (SELECT MAX(age) FROM students)