CSE 311L Week 4 SQL Commands Summary

Displaying Data from Multiple Tables

Cartesian Product:

```
SELECT last_name, department_name dept_name
FROM emps, depts;
```

Equijoin:

```
SELECT e.employee_id, e.last_name, e.department_id,
    d.department_id, d.location_id
FROM emps e, depts d
WHERE e.department_id = d.department_id;
```

Self-Join (Joining a table to itself):

```
SELECT worker.last_name || ' works for ' || manager.last_name
FROM emps worker, emps manager
WHERE worker.manager_id = manager.employee_id;
```

Join using ON clause:

```
SELECT e.employee_id, e.last_name, e.department_id,
    d.department_id, d.location_id
FROM emps e
JOIN depts d ON (e.department_id = d.department_id);
```

Three-Way Join:

```
SELECT employee_id, city, department_name
FROM emps e
JOIN depts d ON d.department_id = e.department_id
JOIN locs l ON d.location_id = l.location_id;
```

Outer Joins

LEFT OUTER JOIN:

```
SELECT e.last_name, e.department_id, d.department_name
FROM emps e
LEFT OUTER JOIN depts d ON (e.department_id = d.department_id);
```

RIGHT OUTER JOIN:

```
SELECT e.last_name, e.department_id, d.department_name
FROM emps e
RIGHT OUTER JOIN depts d ON (e.department_id = d.department_id);
```

FULL OUTER JOIN:

```
SELECT e.last_name, e.department_id, d.department_name
FROM emps e
FULL OUTER JOIN depts d ON (e.department_id = d.department_id);
```

Join with Additional Condition:

```
SELECT e.employee_id, e.last_name, e.department_id,
    d.department_id, d.location_id
FROM emps e
JOIN depts d ON (e.department_id = d.department_id)
```

```
AND e.manager_id = 149;
```

Group Functions

AVG, MAX, MIN, SUM:

```
SELECT AVG(salary), MAX(salary), MIN(salary), SUM(salary)
FROM emps
WHERE job_id LIKE '%REP%';
```

COUNT:

```
SELECT COUNT(DISTINCT department_id)
FROM emps;
```

GROUP BY Clause

Single Column:

```
SELECT department_id, AVG(salary)
FROM emps
GROUP BY department_id;
```

Multiple Columns:

```
SELECT department_id dept_id, job_id, SUM(salary)
FROM emps
GROUP BY department_id, job_id;
```

HAVING Clause:

```
SELECT job_id, SUM(salary) PAYROLL
```

```
FROM emps
WHERE job_id NOT LIKE '%REP%'
GROUP BY job_id
HAVING SUM(salary) > 13000
ORDER BY SUM(salary);
```

Nested Group Functions:

```
SELECT MAX(AVG(salary))
FROM emps
GROUP BY department_id;
```