Week 5 Notes

Question from last week about comparison operators

=, >, <, <=, >=, <> (!=)

Distinct

```
SELECT cust_city FROM customer;
vs
Select DISTINCT custCity FROM customer;
```

Nulls

WHERE description IS NOT NULL WHERE description IS NULL

Ordering

```
SELECT *
FROM product
ORDER BY price desc;
```

Aggregate Functions

- Count

```
SELECT count(*)
FROM product
WHERE price = 300;
```

- Sum (SUM)
- Average (AVG)
    - When calculating an average rows with a null value for that attribute are not counted
- Min (MIN)
- Max (MAX)

```
SELECT MIN(price), MAX(price),
       AVG(price), SUM(price)
FROM product;
```

Insertions Using Queries

```
insert into brisbane_customer
select *
from customer
where custcity ='Brisbane';
```

> This is under the assumption that both tables have the same structure

Grouping in SQL

Break values into different groups based on the value of an attribute

```
SELECT cCity, count(*)
FROM customer
GROUP BY cCity;
```

> For example counting how many customers are from each city

- Use GROUP BY clause to get sub-totals

Having

Listing groups that meet some conditions

```
SELECT cCity, count(cNo)
FROM customer
GROUP BY cCity
Having count(cNo) >= 2;
```

> Show customer counts for cities that have 2 or more customers

Where and having combination

- System evaluates the where first, then divide into groups

Selecting from Multiple Tables

- Generally the two tables will have a foreign key relationship
- Give each table an alias [Don't have to use it] **Example Join**

```
SELECT c.cNo, cName, cStreet, cCity
from customer c, orders o
WHERE c.cNo=o.cNo AND pNo=100;
```

Nesting Queries

```
SELECT pNo, price
FROM product
WHERE price > (Select AVG(price)
                FROM product);
```

Group By

```
SELECT pNo, SUM(quantity)
FROM orders
WHERE to_date(ordDate) >= '01-jan-2016'
    AND to_date(ordDate) <'01-jan-2017'
GROUP BY pNo;
```

```
SELECT pNo, SUM(quantity)
FROM orders
WHERE to_date(ordDate) >= '01-jan-2003'
GROUP BY pNo;
```

Joins

- Outer Joins | Union between two tables | Oracle uses a (+) sign