Handout - T-SQL

1) **Choosing all fields (columns)**

SELECT *
FROM *table_name;*

SELECT *
FROM Customer;

2) **Choosing a selected list of fields (columns)**

SELECT column_name [,column_name, …]
FROM table_name;

SELECT f_name, l_name, date_of_birth
FROM Customer;

- The order in which you list the columns affects the way in which they are presented in the resulting output.
- Items within [ ] are optional.

3) **Temporarily renaming columns in query results**

SELECT column_name AS column_heading [,column_name AS column_heading]
FROM table_name;

Example:

SELECT f_name as "Name"
FROM Customer;

4) **Including calculated columns in the results**

SELECT date_due, rate, principal, rate * principal
FROM loan;

- If necessary, use parentheses to clarify order of precedence in a computation, as in a * (b+c)

5) **Eliminating duplicate query results with *distinct***

If you use the keyword *distinct* after the keyword SELECT, you will only get unique rows.
Example:

SELECT rate,
FROM Loan;

VS.

SELECT distinct rate
FROM Loan;

6) **Selecting rows: the *where* clause**

SELECT Select_list
FROM table

WHERE search_conditions;

Example:

```
SELECT *
FROM Customer
WHERE f_name = 'Carl';
```

- In SQL, string are delimited by single quotes, as in 'Carl'

Available Search Conditions Operators

- Comparison operators (=, <, >, !=. <>, <= ,>=)

```
SELECT * FROM loan
WHERE principal > 100000000;
```

- Ranges (**between** and **not between**; inclusive of the end values)

```
SELECT * FROM loan
WHERE rate BETWEEN 7.5 AND 8.5;
```

- Lists (**in** and **not in**)

```
SELECT *
FROM Customer
WHERE city IN ('Cville', 'Roanoke', 'Lexington');
```

- Character matches (**like** and **not like**)

```
SELECT f_name, l_name
FROM Customer
WHERE l_name LIKE 'Fos%';

SELECT f_name, l_name
FROM Customer
WHERE l_name LIKE '_oster';
```

  - "%" (matches any string of zero or more characters) and "_" (matches any one character). In addition to those, brackets can be used to include either ranges or sets of characters.

- Combinations of previous options using logical operators **and**, **or**, and **not**

```
SELECT f_name, l_name
FROM Customer
WHERE l_name LIKE 'Fos%' AND City NOT IN ('Austin', 'Dallas');
```