Aggregate functions

- Applied either to an attribute or an expression
  - MIN(), MAX()
  - SUM(), AVG()
  - COUNT()
- Cannot be used (directly) in a WHERE clause

Ex: List the minimum, maximum, and average credit limit given to the employees.

Aggregate functions: COUNT

- Two options:
  - COUNT(attribute) (or COUNT(expression))
    - Returns the number of non-NULL values (ignores NULLs)
  - COUNT(*)
    - Returns the total number of rows in the resultset

Ex: Count the number of employees hired after 1996;
Count the total number of managers (?)
**COUNT (cont.)**

- MySQL supports `COUNT(DISTINCT attribute)`
  - ...although MS Access does not (!)
- To replicate this behavior, you can also use a subquery:
  - A SELECT always returns a 2D table of values (resultset), or returns a single value (as with an aggregate function)
  - A normal SELECT pulls data from a DB table (relation)
  - A SELECT based on a subquery pulls data from a resultset, or compares data to a single SELECTed value

---

**Subquery**

```
SELECT [column_names] FROM [table_name | (SELECT ...)]
WHERE [criteria <=> (SELECT...)]
```

**Ex:**

1. Count the total number of managers
2. List the prices of the foods that cost less than the average price of the items on the menu, and sort them in ascending order

---

**GROUP BY**

- Used with aggregate functions
- Combines similar records and into a single record
- You **must** GROUP BY every SELECT-ed field that is not part of some aggregate function

```
SELECT Item_Name, SUM(Quantity*Price) as Sales FROM SalesItem GROUP BY Item_Name ORDER BY Item_Name ASC
```
SELECT statement format (again)

SELECT [DISTINCT | ALL] {* | column1 [AS new_name] [, ...]}
FROM table_name [alias] [, ...]
[WHERE condition(s)]
[GROUP BY column_list] [HAVING condition]
[ORDER BY column_list]

GROUP BY (Cont.)

Ex: For each Dept_Code, list:

(1) The number of employees (as count) and the total credit limit (as total)

(2) Each manager ID in ASC order, the number of people they supervise in DESC order and the total credit limit for those employees

HAVING

• The HAVING clause acts as a filter on the output of a GROUP BY clause
  – Allows you to restrict results from aggregate functions

Ex: For each manager who supervises more than 2 employees, list the number of people they supervise with their total credit limit.
Concatenation / String manipulation

- The CONCAT( ) operator is another example of an operator for manipulating output data
  - MS Access uses the & character instead

```sql
SELECT CONCAT(X, Y, Z) AS [new output]
FROM...
```

**Ex:** List each employee’s first name, last name then ‘works in the’ Dept_Code and ‘dept’. Name the Column "Employee Assignments"

---

Action queries

- INSERT
- UPDATE
- DELETE

---

**INSERT statement**

- Insert a single record into a table

```sql
INSERT INTO TableName
  ([Column1, Column2, ...])
VALUES (Value1 [, Value2,...])
```

- Values must be formatted correctly by data type:
  - (text) ‘Value1’ : (date) ‘Value1’ : (number) Value1

**Ex:** Add a new item into the Item table
UPDATE statement

- Change values within one or more rows in a table

```sql
UPDATE Table Name
SET Column1=Value1 [, Column2=Value2, ...]
WHERE criteria (is true)
```

- Values must be formatted correctly by data type

**Ex:** Increase the credit limit by 20% of all employees who have been with the company at least 10 years

DELETE statement

- Delete one or more rows from a table

```sql
DELETE FROM Table Name
WHERE criteria (is true)
```

- Always deletes entire row(s)

**Ex:** Delete the finance department's data from Department