# Group By

# GROUP BY

- **Used with aggregate functions** to group the result set
- Can divide the table into logical groups or categories and perform calculations for each group

```
SELECT make, COUNT(*) as numberOf
FROM carInventory
GROUP BY make;
```

| make | numberOf |
|---|---|
| Acura | 21 |
| Aptera | 1 |
| Aston Martin | 5 |
| Audi | 25 |
| Bentley | 12 |
| BMW | 37 |

# GROUP BY

- The GROUP BY clause is used for aggregation of a result set
- The database returns rows of information that is filtered by the where clause – this result is set is then grouped by the values of one or more columns.
- The GROUP BY lists the columns that are used to determine the groups
- The returned (grouped) rows consists of the information from one or more rows – these are produced rows
- A grouped row is a new row representing each group of rows found during aggregation

# GROUP BY

```
SELECT make, SUM(price),COUNT(*) as numberOf
FROM carInventory
GROUP BY make;
```

| make | SUM(price) | numberOf |
|---|---|---|
| Acura | 689700.13 | 21 |
| Aptera | 25580.07 | 1 |
| Aston Martin | 176896.22 | 5 |
| Audi | 743231.61 | 25 |
| Bentley | 315743.38 | 12 |
| BMW | 990397.44 | 37 |

# GROUP BY

- Gathers all of the information from the FROM clause
- Filters the information by the WHERE clause (if present)
- Aggregates the remaining rows into groups.
- Only group rows can be used in the SELECT clause

# GROUP BY

- The GROUP BY clause comes **AFTER** the WHERE clause and **BEFORE** the ORDER BY clause
- No columns can appear in the SELECT clause UNLESS they are also included in the GROUP BY clause

```
SELECT make, modelYear, COUNT(*) as numberOf
FROM carInventory
GROUP BY make;
```

```
SELECT make, modelYear, COUNT(*) as numberOf
FROM carInventory
GROUP BY make, modelYear;
```