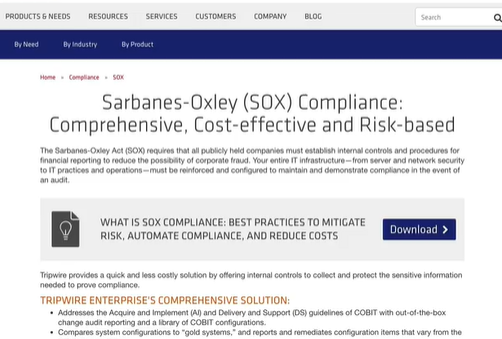
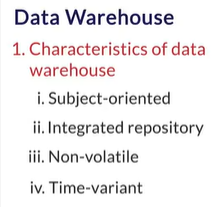
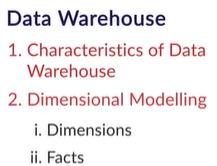
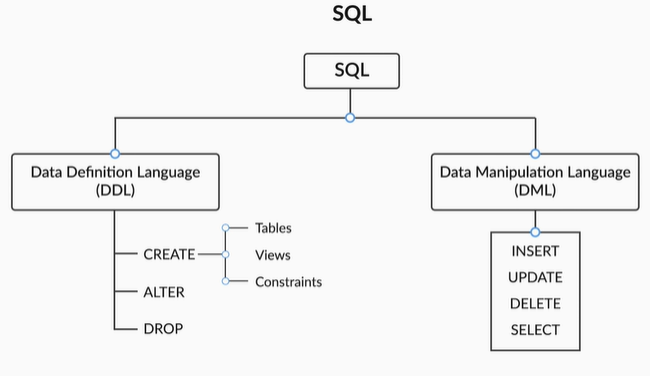
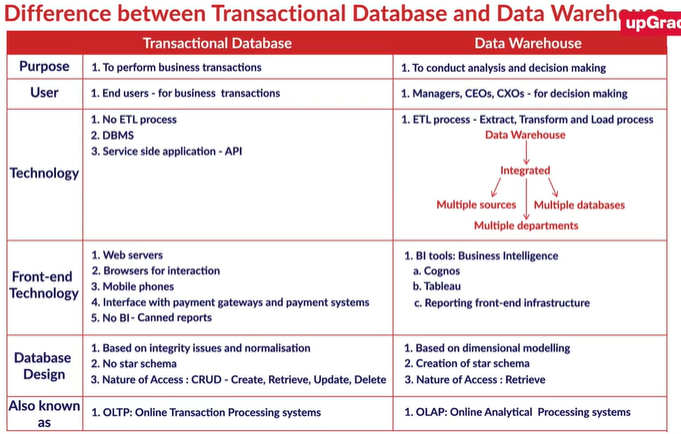
**OLAP**, which stands for **Online Analytical Processing systems**. OLAP is used to extract business-relevant information and perform analysis on the data stored in data warehouses.

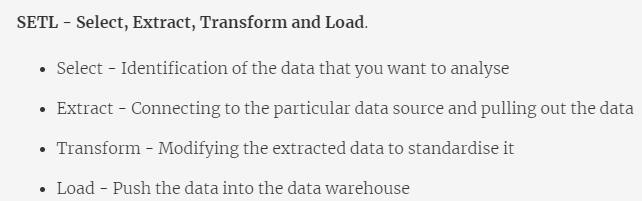
 SOX Standard

A data warehouse is a collection of data. It demonstrates the following properties:

* **Subject-oriented:** A data warehouse should contain information about a few well-defined subjects rather than the enterprise.
* **Integrated:** A data warehouse is an integrated repository of data. It contains information from various systems within an organisation.
* **Non-volatile:** The data values in a database cannot be changed without a valid reason.
* **Time-variant:** A data warehouse contains historical data for analysis.

Basically, facts are the numerical data in a data warehouse and dimensions are the metadata (that is, data explaining some other data) attached to the fact variables. Both facts and dimensions are equally important for generating actionable insights from a data set. 

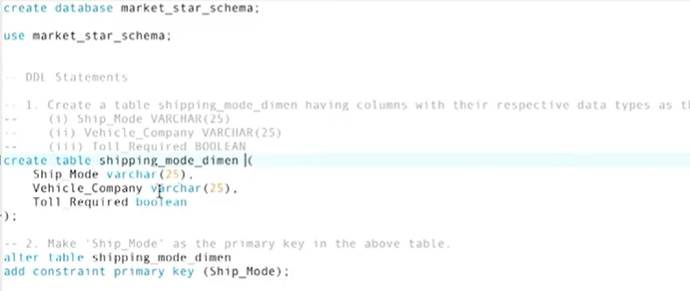


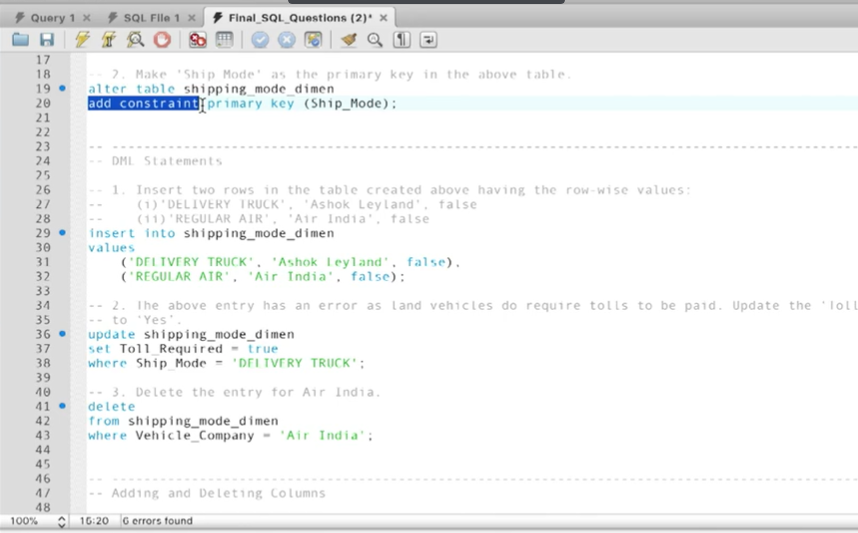


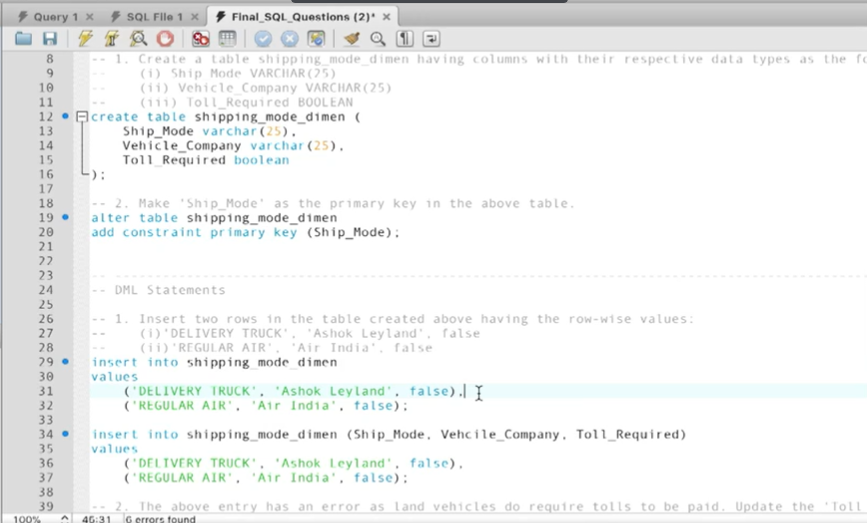
Entity Constraints, Referential Constraints, Semantic Constraints

As you learnt in the video, the constituents of an ERD are as follows:

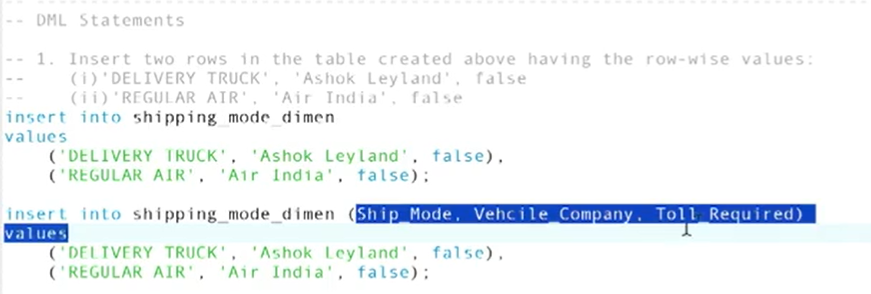
* **Entity Type/Entity**: It is nothing but a table in the schema. For example, 'orders' and 'payments' are both entity types.
* **Attribute**: It is a column in an entity type. For example, 'orderNumber' is an attribute in the 'orders' entity type.
* **Relationship Types**: They are the lines between tables, which define the relationships among them. These can be of various types based on their cardinalities, i.e., **one-to-one**, **one-to-many**, **many-to-many**, etc.

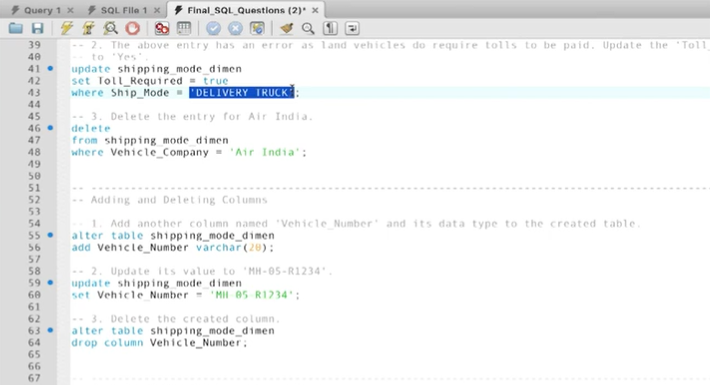


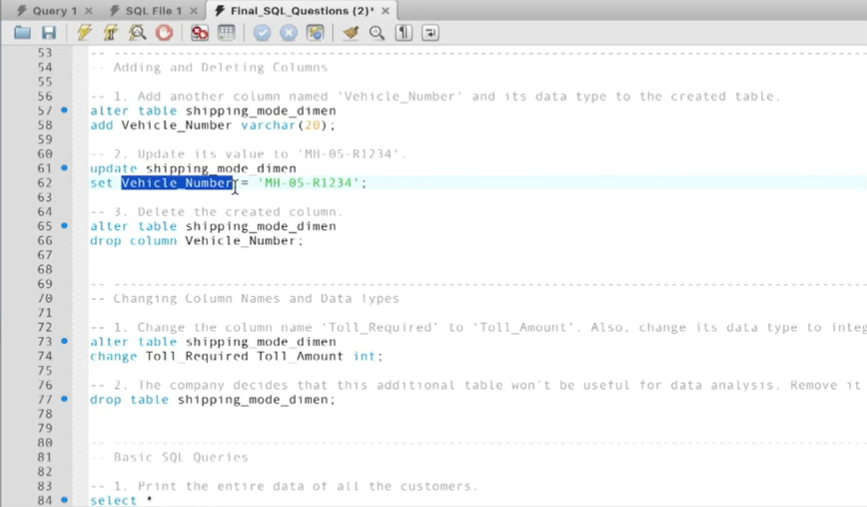










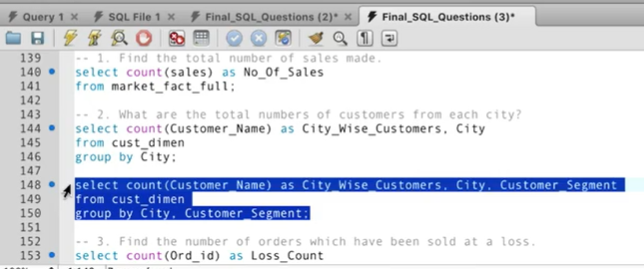
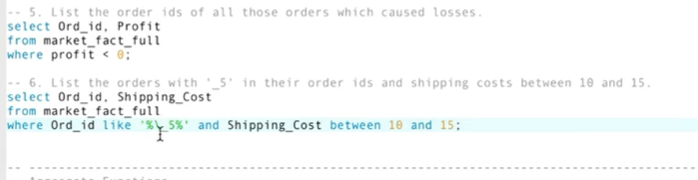


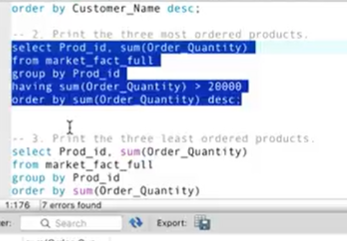
The DDL statements are:

* CREATE
* ALTER
* DROP

The DML statements are:

* INSERT
* UPDATE
* DELETE
* SELECT
* While Data Definition Language statements are used to change the structure of the database, Data Manipulation Language statements are used to change the data itself inside the database.
* Now that you are ready to use MySQL for its primary intended purpose, that is, data analysis, it's time to write useful queries to derive actionable insights from the sea of data around you.



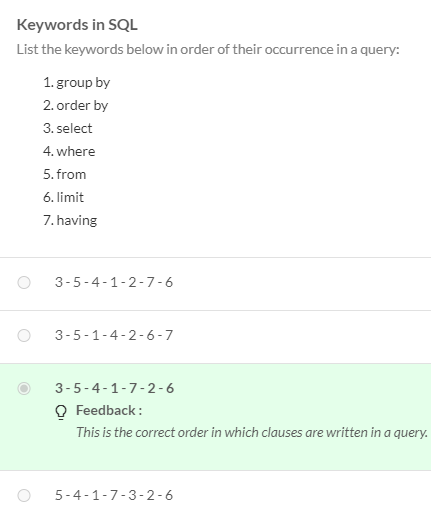


If you want to filter rows of data, use the where clause, if you want to filter after applying aggregate function, use the having clause

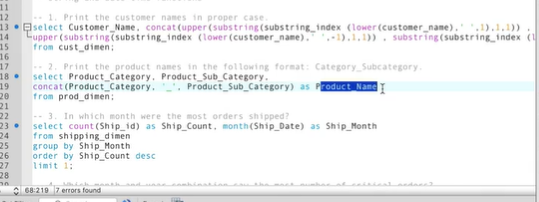
The 'having'clause is typically used when you have to apply a filter condition on an 'aggregated value'.This is because the 'where' clause is applied before aggregation takes place, and thus, it is not useful when you want to apply a filter on an aggregated value.

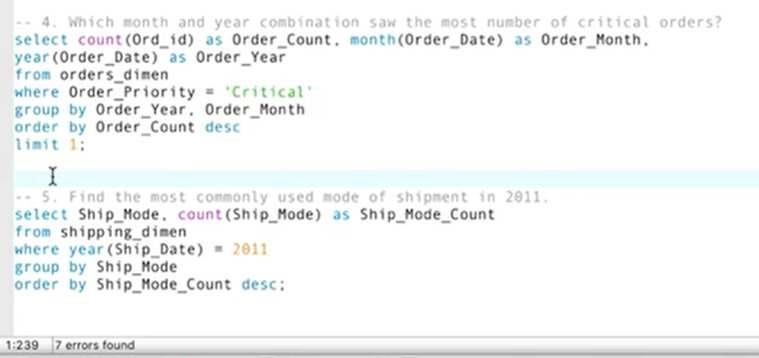
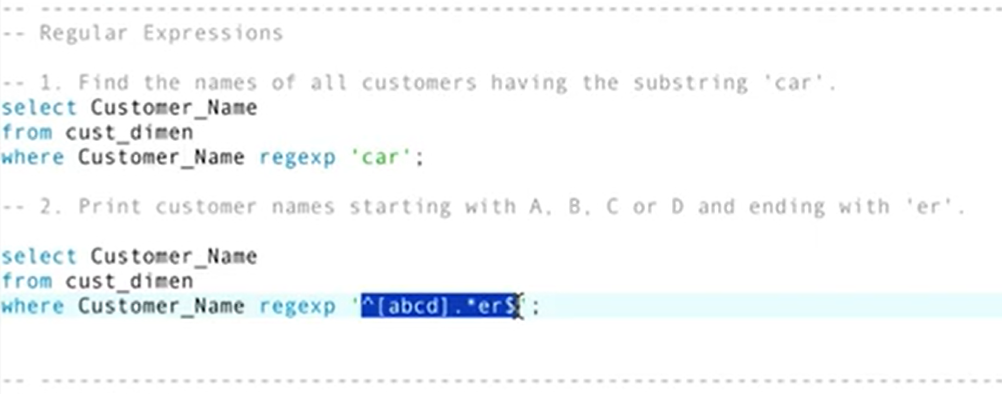
In other words, the 'having'clause is equivalent to a 'where' clause after the 'group by' clause has been executed but before the 'select' clause is executed. You can read[this StackOverflow answer](http://stackoverflow.com/questions/15767794/sql-how-to-filter-after-aggregation) to understand the 'having' clause better.

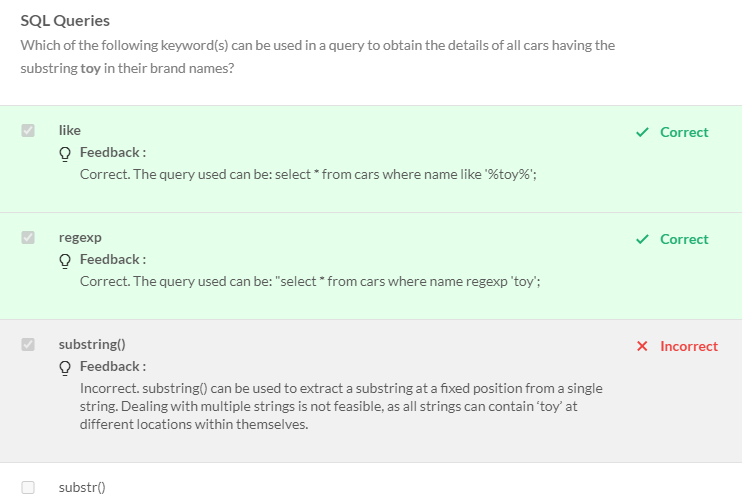
It is important to not get confused between the 'having' and 'where' clauses. For example, if you want to display the list of all employees having salary >= 30,000, then you can use the 'where' clause since there is no aggregation taking place in this query. But if you want to display the list of all employees having salary <= the average salary, where avg() is the aggregation function, you'll have to use the 'having' clause.

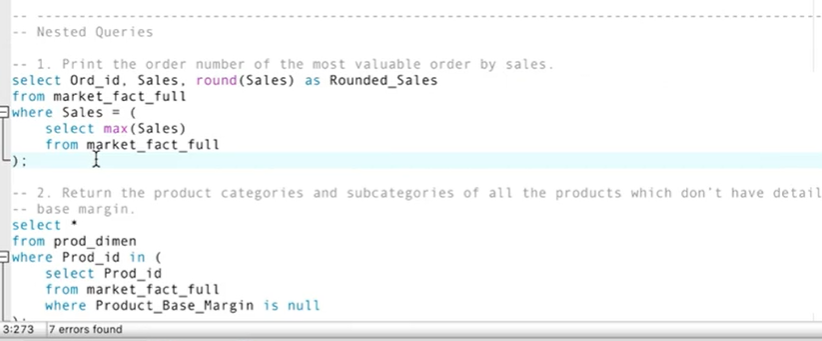


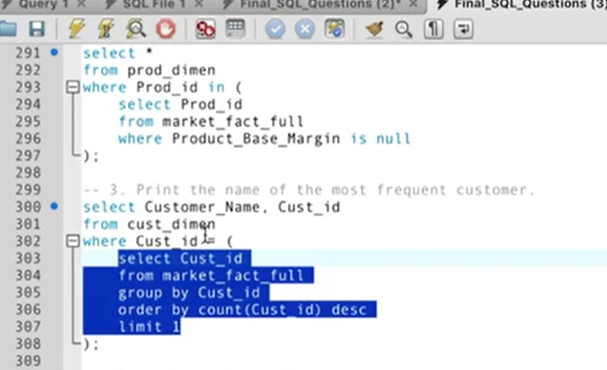
Select, from, where, group by, having order by, limit



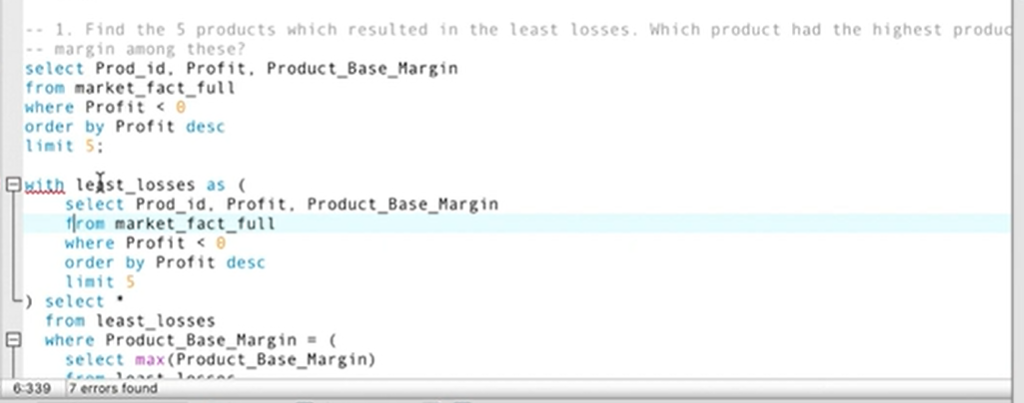


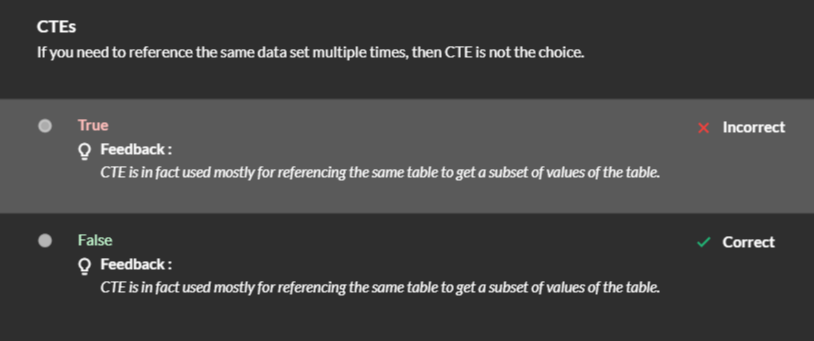






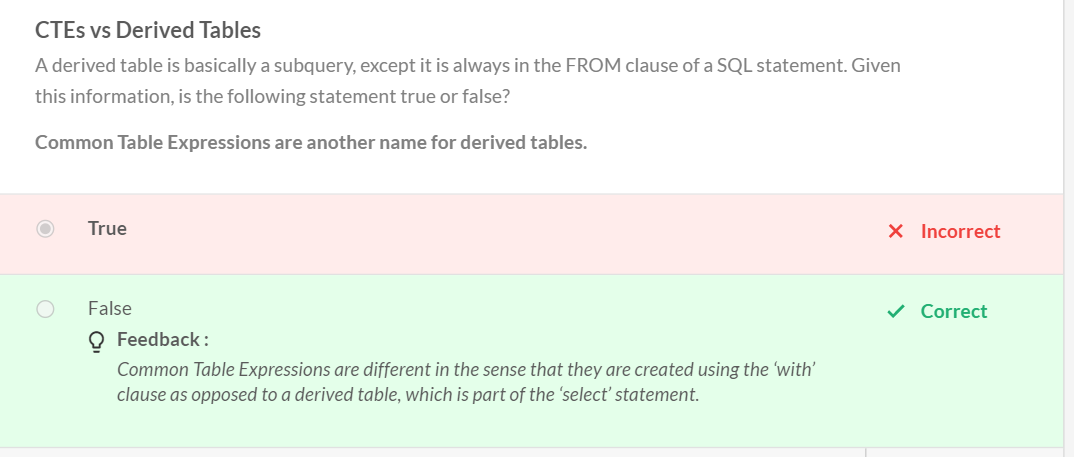
CTE – Common Table Expression

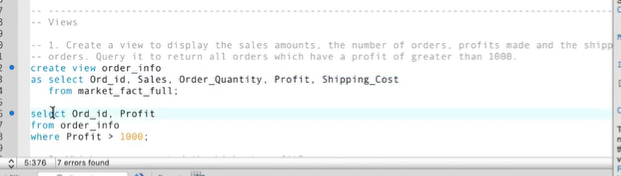






A Common Table expression is used to create a temporary table which is smaller than the existing table. This smaller table cannot be individually queried, that returns an error. The CTE has to be used as part of the main query.





Note that it is not necessary that views would always be preferred to CTEs. When you know for sure that you need to subset data from a table only once, you should use a CTE then to avoid extra memory and space usage.

Now, you learnt that a complete query (in a sense) has the following basic outline:

**select (attributes)**

**from (table)**

**where (filter\_condition)**

**group by (attributes\_to\_be\_grouped\_upon)**

**having (filter\_condition\_on\_grouped\_values)**

**order by (values)**

**limit (no\_of\_values\_to\_display);**

You also learnt the following important concepts:

* Relational, arithmetic and logical operators
* Aggregate functions
* Regular expressions
* Nested queries
* Common Table Expressions
* Views
* Advantages of views over CTEs

In the next session, you will finally learn all about joins and set operations.

