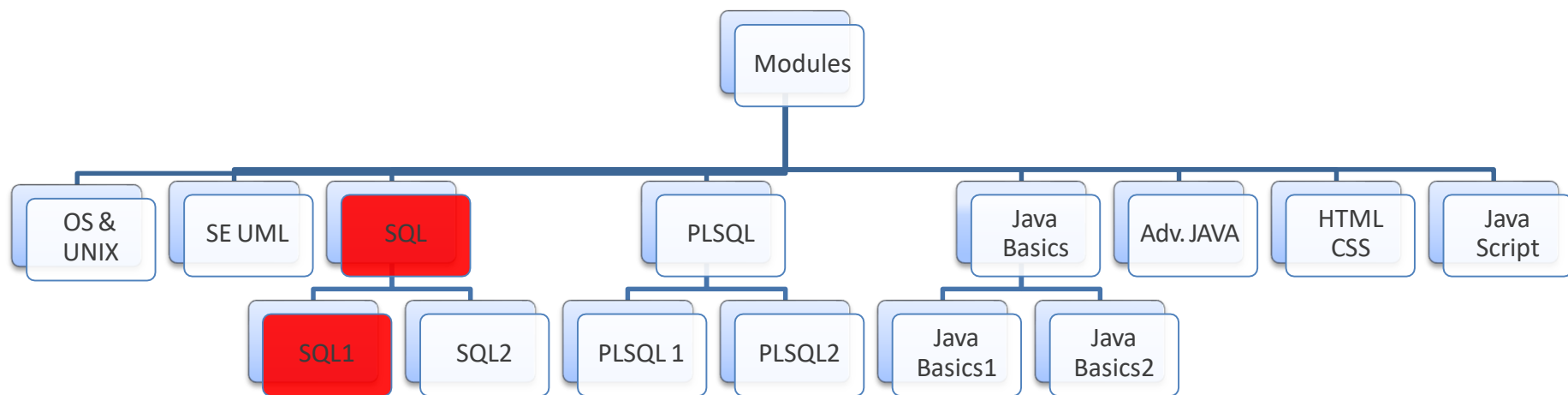


SQL-Part 1

Module Overview

The following module hierarchy presents the technical modules required to build the basic IT skills and acquaints you with relevant technology basics.

The current module – **SQL 1** (highlighted in red) underwrites Basics of SQL 1 and will enable you to enhance one's query writing skills.



* **Recommended duration: 6 hours**

Module Objectives

By the end of this module, you will be able to:

- Define RDBMS Concepts
- Draw ER Diagrams
- Normalize the data using Normal Forms
- Retrieve data using DQL statements (SELECT Statement)
- Write sub queries
- Join tables to retrieve data from multiple tables

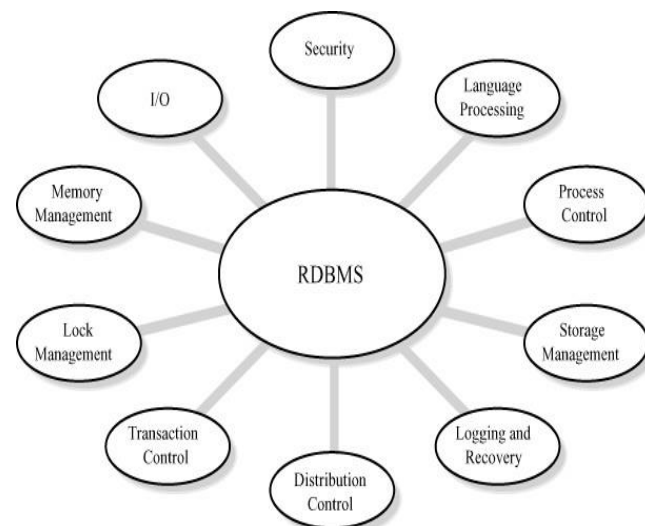
RDBMS - Introduction

What is RDBMS ?

- RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.
- A Relational Database Management System (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

What is Table?

- The data in RDBMS is stored in database objects called tables. The table is a collection of related data entries and it consists of columns and rows.
- Remember, a table is the most common and simplest form of data storage in a relational database.



References

- <http://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm>

RDBMS - ER Diagram

ER

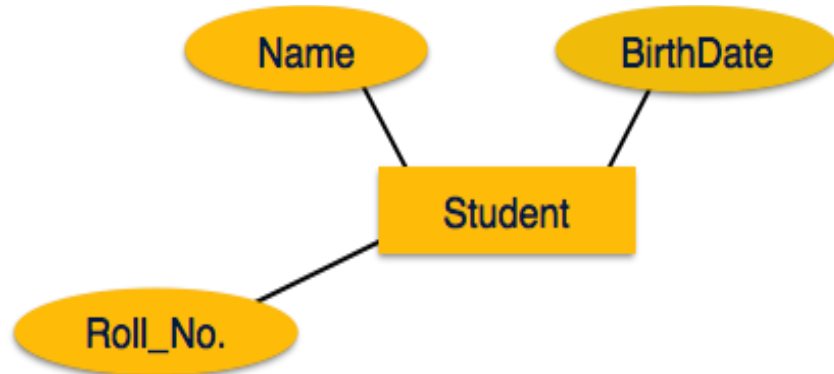
- Entity Relationship Diagram is a visual representation of data that describes how data is related to each other.

Entity

- Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

Attributes

- Attributes are properties of entities. Attributes are represented by means of eclipses. Every eclipse represents one attribute and is directly connected to its entity (rectangle).



References

- http://www.tutorialspoint.com/dbms/er_diagram_representation.htm

RDBMS - Normalization

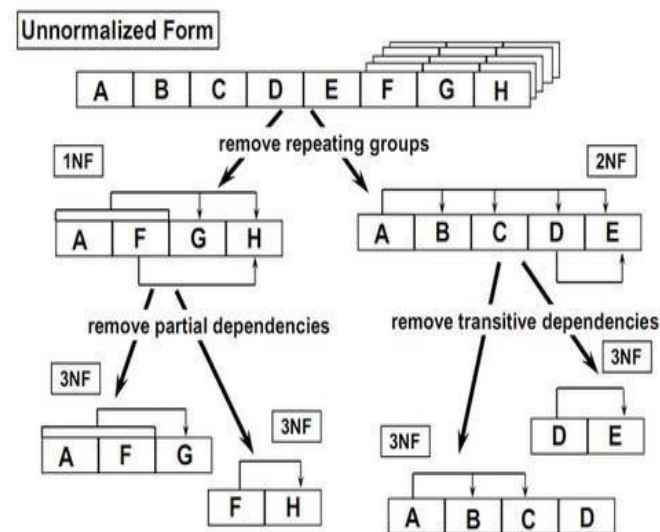
Database Normalization, or Data Normalization, is a technique to organize the contents of the tables for transactional databases and data warehouses.

Normalization is part of successful database design; without normalization, database systems can be inaccurate, slow, and inefficient, and they might not produce the data you expect.

Data Normalization Rules:

Level	Rule
First normal form(1NF)	An entity type is in 1NF when it contains no repeating groups of data.
Second normal form(2NF)	An entity type is in 2NF when it is in 1NF and when all of its non-key attributes are fully dependent on its primary key
Third normal form(3NF)	An entity type is in 3NF when it is in 2NF and when all of its attributes are directly dependent on the primary key

Normalization Process



References

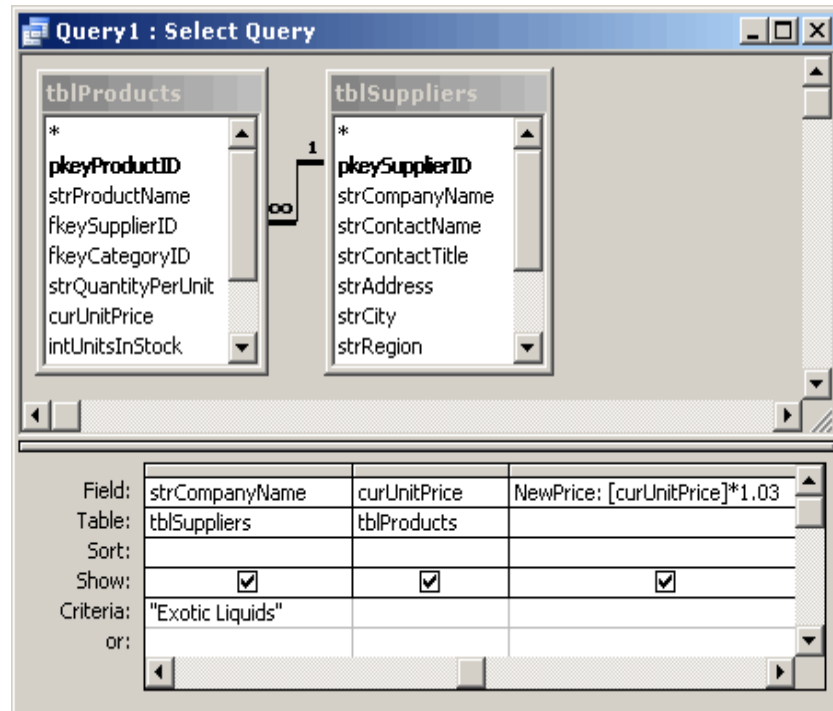
- http://www.tutorialspoint.com/dbms/database_normalization.htm

RDBMS - Querying Database

- SQL – SELECT Query
- SQL SELECT statement is used to fetch the data from a database table which returns data in the form of result table. These result tables are called result-sets.

Syntax:

- The basic syntax of SELECT statement is as follows:
- SELECT column1, column2, columnN FROM table_name



References

- <http://www.tutorialspoint.com/sql/sql-select-query.htm>

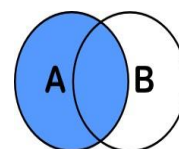
RDBMS - SQL Joins

- The SQL Joins clause is used to combine records from two or more tables in a database.
- A **JOIN** is a means for combining fields from two tables by using values common to each.

SQL Join Types:

There are different types of Joins available in SQL:

- Inner Join:** Returns rows when there is a match in both tables
- Left Join:** Returns all rows from the left table, even if there are no matches in the right table
- Right Join:** Returns all rows from the right table, even if there are no matches in the left table.
- Full Join:** Returns rows when there is a match in one of the tables
- Self Join:** Used to join a table to itself as if the table were two tables, temporarily renaming at least one table in the SQL statement

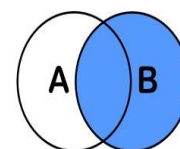


```
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
```

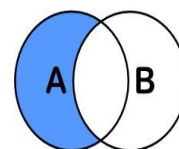


CHEATSHEET
SQL JOINS

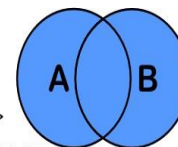
```
SELECT <auswahl>
FROM tabelleA A
INNER JOIN tabelleB B
ON A.key = B.key
```



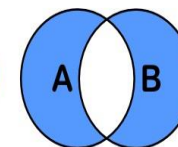
```
SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key
```



```
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
WHERE B.key IS NULL
```



```
SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key
```



```
SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL
```

```
SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL
```

References

- <http://www.tutorialspoint.com/sql/sql-using-joins.htm>

Additional References

To explore more on the subject, refer the below links and books:

Links :-

<http://www.tutorialspoint.com/dbms/>

<http://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm>

SQL Reference:

https://docs.oracle.com/cd/E11882_01/server.112/e41084.pdf

Self Check?

Instructions to write Self Evaluation Sheet:

Open the excel sheet, refer SQL Part 1 sheet, write down the solutions for all questions, save a local copy in your machine.

Lab Assignment

- Refer ***Assignment Document*** to complete the tasks on the required timeline
- You are required to submit the Solutions for the given assignment and refer the ***Participant guide*** to get know the submission procedure.

Module Summary

Now that you have completed this module, you will be able to:

- Apply RDBMS Concepts
- Design ER Diagrams
- Normalize the data available in the database using Normal Forms
- Obtain the data using DQL statements (SELECT Statement)
- Use sub queries in select statement.
- Combine tables together to retrieve data from multiple tables.

Thank you!