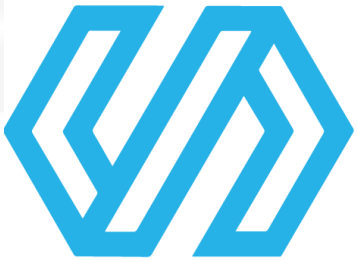
Databases and SQL



SKILLO

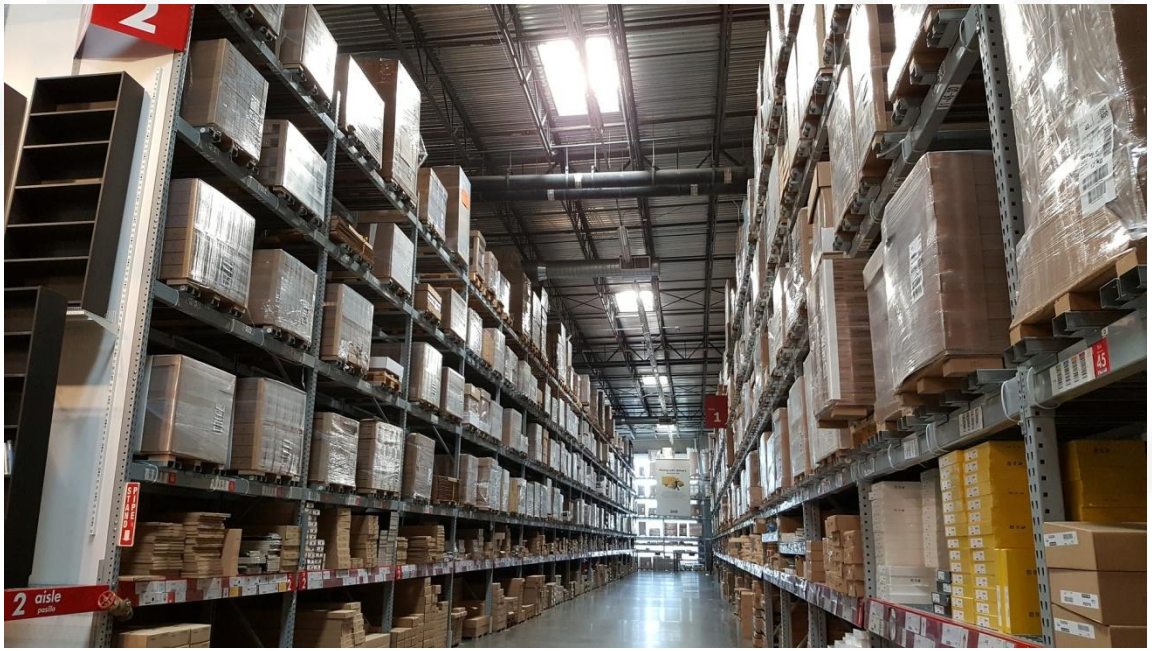
Agenda

1. What is database?
2. DBMS
3. Popular DBMS
4. Database types
5. Relational DB model
6. NoSQL DB model
7. Basic relational database concepts

Agenda

8. Data types
9. What is SQL?
10. SQL elements
11. DDL – Data Definition Language
12. DML – Data Manipulation Language
13. Join Clause
14. Group by and Aggregate functions

What is database?



A database is a collection of information that is organized so that it can be easily accessed, managed and updated.

SKILLO

DBMS

Database-management system (**DBMS**) is a computer-software application that interacts with users, other applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases.

Popular DBMS

MS SQL Server

Oracle

MySQL

PostgreSQL

SQLite

IBM DB2

Teradata

Database types

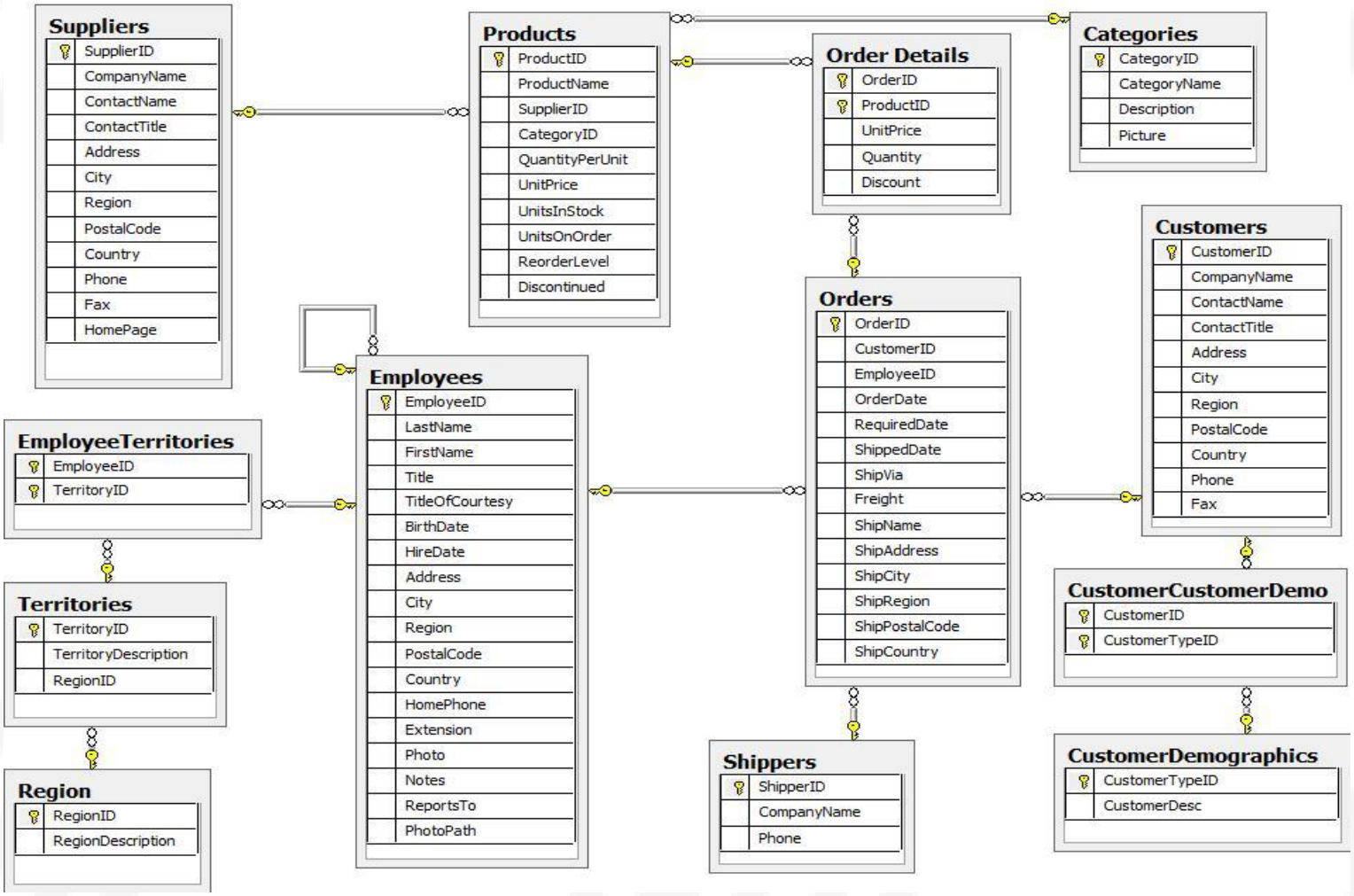
Relational databases

Flat-File Database

NoSQL

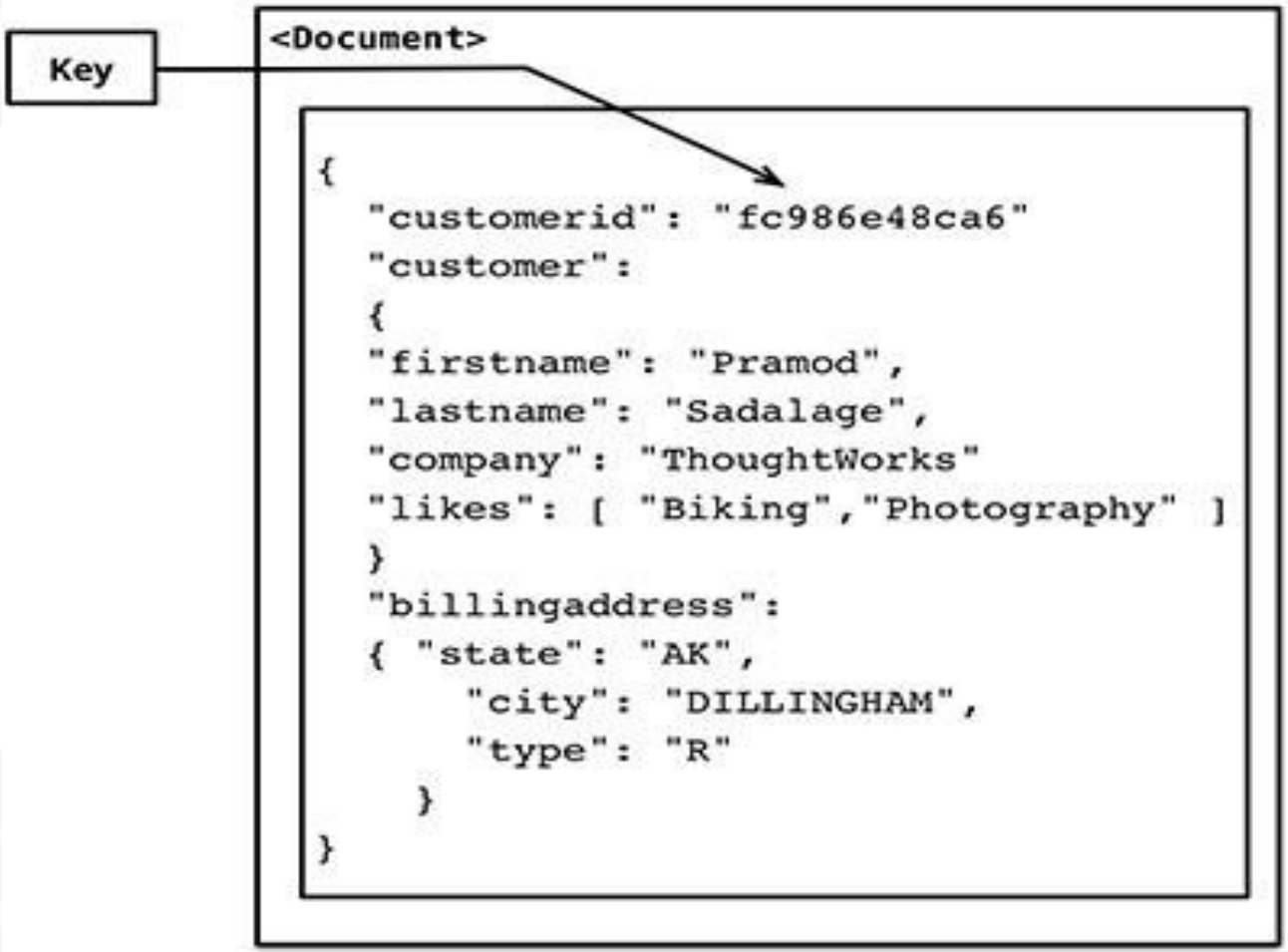
Etc.

Relational DB model



SKILLO

NoSQL DB Model



SKILLO

Basic relation DB concepts

Table - A table is a collection of data held in a structured format within a database. It consists of columns, and rows.

Basic relation DB concepts

Columns stores a specific data type

Row →
Or record

Emp No	Name	Age	Department	Salary
001	Alex S	26	Store	5000
002	Golith K	32	Marketing	5600
003	Rabin R	31	Marketing	5600
004	Jons	26	Security	5100

SKILLO

Basic relation DB concepts

Relationship types

1. one-to-one **2.** many-to-one **3.** one-to-many **4.** many-to-many

Primary key - uniquely identifies each record in a database table. Each table should have a primary key, and each table can have only ONE primary key. A primary key column cannot contain NULL values. Primary key can be built from more than one column.

Foreign key - points to a PRIMARY KEY in another table. One table can have MORE than ONE foreign keys. Foreign key can contain NULL value.

NULL value - indicate that a data value does not exist in the database

Basic relation DB concepts

Relationship types

1. one-to-one **2.** many-to-one **3.** one-to-many **4.** many-to-many

Primary key - uniquely identifies each record in a database table. Each table should have a primary key, and each table can have only ONE primary key. A primary key column cannot contain NULL values. Primary key can be built from more than one column.

Foreign key - points to a PRIMARY KEY in another table. One table can have MORE than ONE foreign keys. Foreign key can contain NULL value.

NULL value - indicate that a data value does not exist in the database

Data types

- ✓INT
- ✓DECIMAL(p,s)
- ✓CHAR(n)
- ✓VARCHAR(n)
- ✓BOOLEAN
- ✓DATE
- ✓TIME
- ✓TIMESTAMP

SKILLO

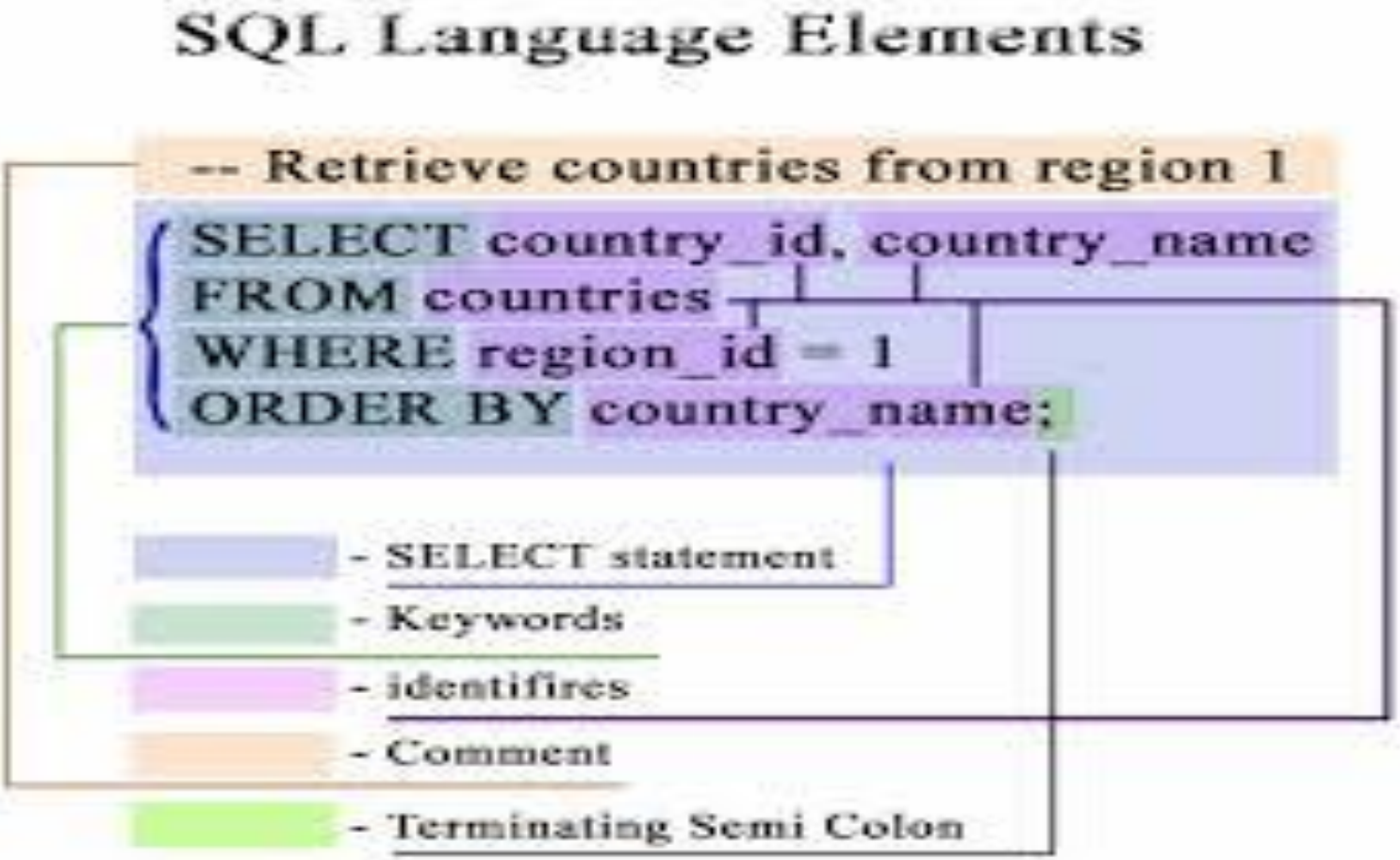
What is SQL?

SQL stands for Structured Query Language

Language for manipulating data in relational databases (DB)

Minor differences in implementations between SQL vendors

SQL elements



SKILLO

DDL – Data Definition Language

- ✓ CREATE
- ✓ DROP
- ✓ TRUNCATE
- ✓ ALTER

SKILLO

CREATE

CREATE TABLE *table_name*
(
column_name1 data_type(*size*),
column_name2 data_type(*size*),
column_name3 data_type(*size*),
....
);

CREATE TABLE Person
(
PersonID int,
LastName varchar(255),
FirstName varchar(255),
Address varchar(255),
City varchar(255)
);

SKILLO

DROP

- DROP TABLE table_name;
- DROP DATABASE database_name;
- DROP TABLE persons;
- DROP DATABASE stores;

SKILLO

TRUNCATE

- Truncate is used only with TABLES

- TRUNCATE TABLE table_name;
- TRUNCATE TABLE persons;

ALTER

- ALTER TABLE table_name
ADD column_name datatype;

- ALTER TABLE table_name
DROP COLUMN column_name;

- ALTER TABLE Persons
ADD DateOfBirth date;

- ALTER TABLE Persons
DROP COLUMN DateOfBirth;

SKILLO

DML – Data Manipulation Language

- ✓ SELECT
- ✓ INSERT
- ✓ UPDATE
- ✓ DELETE

SELECT

- The SELECT statement is used to select data from a database.

- SELECT *column_name,column_name*
FROM *table_name*;

- SELECT * FROM *table_name*;

- SELECT CustomerName, ContactName FROM Customers;

- SELECT * FROM Customers;

SELECT DISTINCT

- The SELECT DISTINCT statement is used to return only distinct (different) values.

- SELECT DISTINCT *column_name,column_name* FROM *table_name*;

- SELECT distinct city FROM Customers;

- SELECT DISTINCT SupplierID, CategoryID FROM Products

SKILLO

WHERE

- The WHERE clause is used to filter records.

- SELECT *column_name, column_name*
FROM *table_name*
WHERE *column_name operator value*;

- SELECT *
FROM *table_name*
WHERE *column_name operator value*;

- SELECT ContactName
FROM [Customers]
WHERE City = 'México D.F.'

- SELECT *
FROM [Customers]
WHERE City = 'México D.F.'

SKILLO

AND/OR operators

- The AND operator displays a record if both the first condition AND the second condition are true.

- The OR operator displays a record if either the first condition OR the second condition is true.

- SELECT *
FROM [Customers]
WHERE City = 'México D.F.'
AND Country = ''Germany‘

- SELECT *
FROM [Customers]
WHERE City = 'México D.F.'
OR Country = 'Germany‘

- SELECT * FROM Customers
WHERE Country='Germany'
AND (City='Berlin' OR City='Aachen');

SKILLO

ORDER BY

- The ORDER BY keyword is used to sort the result-set.

- SELECT *column_name,column_name*
FROM *table_name*
ORDER BY *column_name,column_name* ASC|DESC;

- SELECT * FROM Customers
ORDER BY Country;

- SELECT * FROM Customers
ORDER BY Country DESC;

- SELECT * FROM Customers
ORDER BY Country, CustomerName;

SKILLO

TOP

- The SELECT TOP clause is used to specify the number of records to return.

- MySQL:

```
SELECT *
FROM Persons
LIMIT 5;
```

- ORACLE:

```
SELECT *
FROM Persons
WHERE ROWNUM <=5;
```

- MS SQL Server:

```
SELECT TOP 2 * FROM Customers;
```

SKILLO

LIKE and Wildcards

- The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.
- A wildcard character can be used to substitute for any other character(s) in a string.

Wildcard	Description
%	A substitute for zero or more characters
_	A substitute for a single character
[*charlist*]	Sets and ranges of characters to match
[^*charlist*] or [!*charlist*]	Matches only a character NOT specified within the brackets

IN operator

- The IN operator allows you to specify multiple values in a WHERE clause.

- SELECT *column_name(s)*
  FROM *table_name*
  WHERE *column_name* IN (*value1*,*value2*,...);

- SELECT * FROM Suppliers

WHERE City IN ('Berlin','Tokyo');

SKILLO

BETWEEN operator

- The IN operator allows you to specify multiple values in a WHERE clause.

- SELECT *column_name(s)*
  FROM *table_name*
  WHERE *column_name* BETWEEN *value1* AND *value2;*

- SELECT * FROM Products
  WHERE Price BETWEEN 10 AND 20;

- SELECT * FROM Products
  WHERE Price NOT BETWEEN 10 AND 20;

- SELECT * FROM Products
  WHERE (Price BETWEEN 10 AND 20)
  AND NOT CategoryID IN (1,2,3);

- SELECT * FROM Products
  WHERE ProductName BETWEEN 'C' AND 'M';

INSERT INTO statement

- The INSERT INTO statement is used to insert new records in a table.

- INSERT INTO *table_name*
  VALUES (*value1*,*value2*,*value3*,...);

- INSERT INTO *table_name* (*column1*,*column2*,*column3*,...)
  VALUES (*value1*,*value2*,*value3*,...);

- INSERT INTO *table2*
  SELECT * FROM *table1;*

- INSERT INTO *table2*
  (column_name(s))
  SELECT *column_name(s)*
  FROM *table1;*

SKILLO

INSERT INTO statement

- INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)
  VALUES ('Cardinal','Tom B. Erichsen','Skagen 21','Stavanger','4006','Norway');

- INSERT INTO Customers (CustomerName, City, Country)
  VALUES ('Cardinal', 'Stavanger', 'Norway');

- INSERT INTO Customers (CustomerName, Country)
  SELECT SupplierName, Country FROM Suppliers;

- INSERT INTO Customers (CustomerName, Country)
  SELECT SupplierName, Country FROM Suppliers
  WHERE Country='Germany';

SKILLO

UPDATE statement

- The UPDATE statement is used to update existing records in a table.

- UPDATE *table_name*
  SET *column1=value1,column2=value2,...*
  WHERE *some_column=some_value*;

- UPDATE Customers
  SET ContactName='Alfred Schmidt', City='Hamburg'
  WHERE CustomerName='Alfreds Futterkiste';

!!! DO NOT DO THIS

- UPDATE Customers
  SET ContactName='Alfred Schmidt', City='Hamburg';

SKILLO

DELETE statement

- The DELETE statement is used to delete records in a table.

- DELETE FROM *table_name*
  WHERE *some_column=some_value*;

- DELETE FROM Customers
  WHERE CustomerName='Alfreds Futterkiste' AND ContactName='Maria Anders';

Cannot UNDO DELETE statement. Be careful!!!

- DELETE FROM *table_name*;

  or

  DELETE * FROM *table_name*;

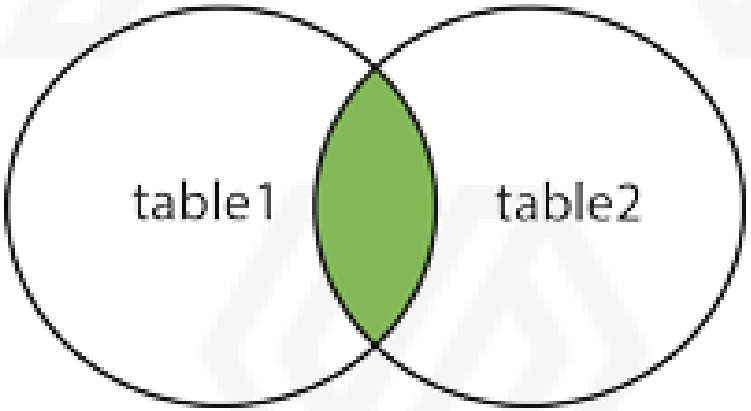
SKILLO

JOIN clause

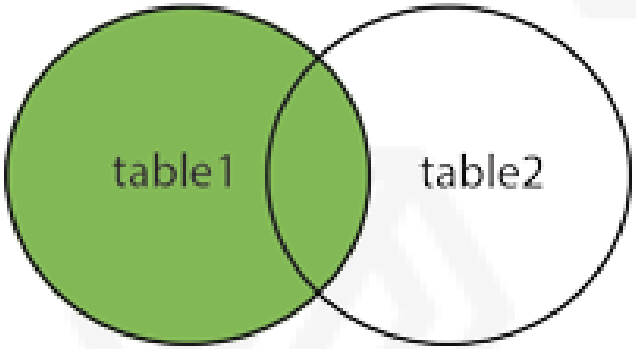
- An SQL JOIN clause is used to combine rows from two or more tables, based on a common field between them.
- JOIN types:
- INNER JOIN - return all rows from multiple tables where the join condition is met
- LEFT JOIN - The LEFT JOIN keyword returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match.
- RIGHT JOIN - The RIGHT JOIN keyword returns all rows from the right table (table2), with the matching rows in the left table (table1). The result is NULL in the left side when there is no match.
- FULL JOIN - The FULL OUTER JOIN keyword returns all rows from the left table (table1) and from the right table (table2).

JOIN clause

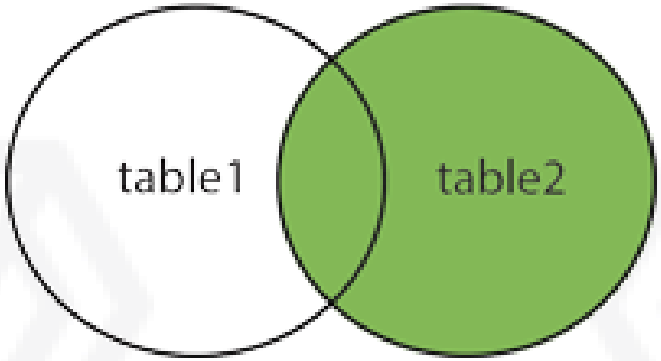
INNER JOIN



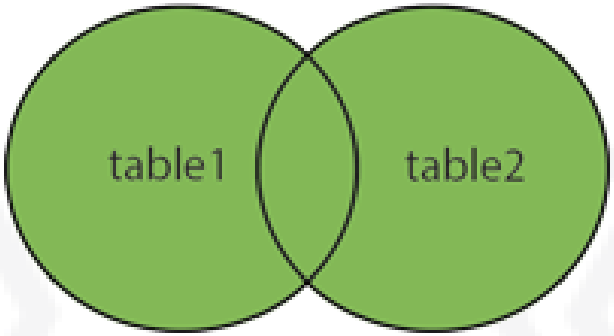
LEFT JOIN



RIGHT JOIN



FULL OUTER JOIN



SKILLO

JOIN clause

- JOIN Syntax:

- SELECT *column_name(s)*
  FROM *table1*
  (INNER/LEFT/RIGHT/FULL OUTER) JOIN *table2*
  ON *table1.column_name=table2.column_name*;

- SELECT Customers.CustomerName, Orders.OrderID
  FROM Customers
  INNER JOIN Orders
  ON Customers.CustomerID=Orders.CustomerID
  ORDER BY Customers.CustomerName;

- SELECT Customers.CustomerName, Orders.OrderID
  FROM Customers
  LEFT JOIN Orders
  ON Customers.CustomerID=Orders.CustomerID
  ORDER BY Customers.CustomerName;

GROUP BY and Aggregate functions

- The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

- SELECT column_name, aggregate_function(column_name)
  FROM table_name
  WHERE column_name operator value
  GROUP BY column_name;

- The following SQL statement counts as orders grouped by shippers:

- SELECT Shippers.ShipperName,COUNT(Orders.OrderID) AS NumberOfOrders
- FROM Orders
  LEFT JOIN Shippers
  ON Orders.ShipperID=Shippers.ShipperID
  GROUP BY ShipperName;

SKILLO

GROUP BY and Aggregate functions

Useful Aggregate functions:

- AVG()
- COUNT()
- SUM()
- MAX()
- MIN()

SKILLO

EXERCISES

Q & A

SKILLO

THANK YOU

SKILLO