IBA

Intro to SQL

CS 341 Database Systems Lab

Relational Database Management System

ORACLE®
DATABASE

Microsoft®
SQL Server®

MySQL®

PostgreSQL

DBMS comparison

	ORACLE	MS SQL Server	MySQL	PostgreSQL	IBM DB2
Supported OS	AIX, HP-UX, Linux, OS X, Solaris, Windows, z/OS.	Linux and Windows.	FreeBSD, Linux, OS X, Solaris, and Windows	FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix and Windows.	AIX®, Linux®, Windows, Mac OS X
Supported Programming Languages	C, C#, C++, Clojure, Cobol, Delphi, Eiffel, Erlang, Fortran, Groovy, Haskell, Java, JavaScript, Lisp, Objective C, OCaml, Perl, PHP, Python, R, Ruby, Scala, Tcl, Visual Basic.	C#, C++, Delphi, Go, Java, JavaScript, PHP, Python, R, Ruby, Visual Basic	Ada, C, C#, C++, D, Delphi, Eiffel, Erlang, Haskell, Java, JavaScript, Objective-C, OCaml, Perl, PHP, Python, Ruby, Scheme, and Tcl	Net, C, C++, Delphi, Java, JavaScript, Perl, PHP, Python, and Tcl	C, C++, COBOL Fortran, Java™ Perl, PHP, Python Ruby/Ruby on Rails, REXX, C#, VB .NET and other .NET languages

Intro to Software

- Oracle 19c or 21c

(21c is newer, 19c is more stable)

- SQL Developer as IDE

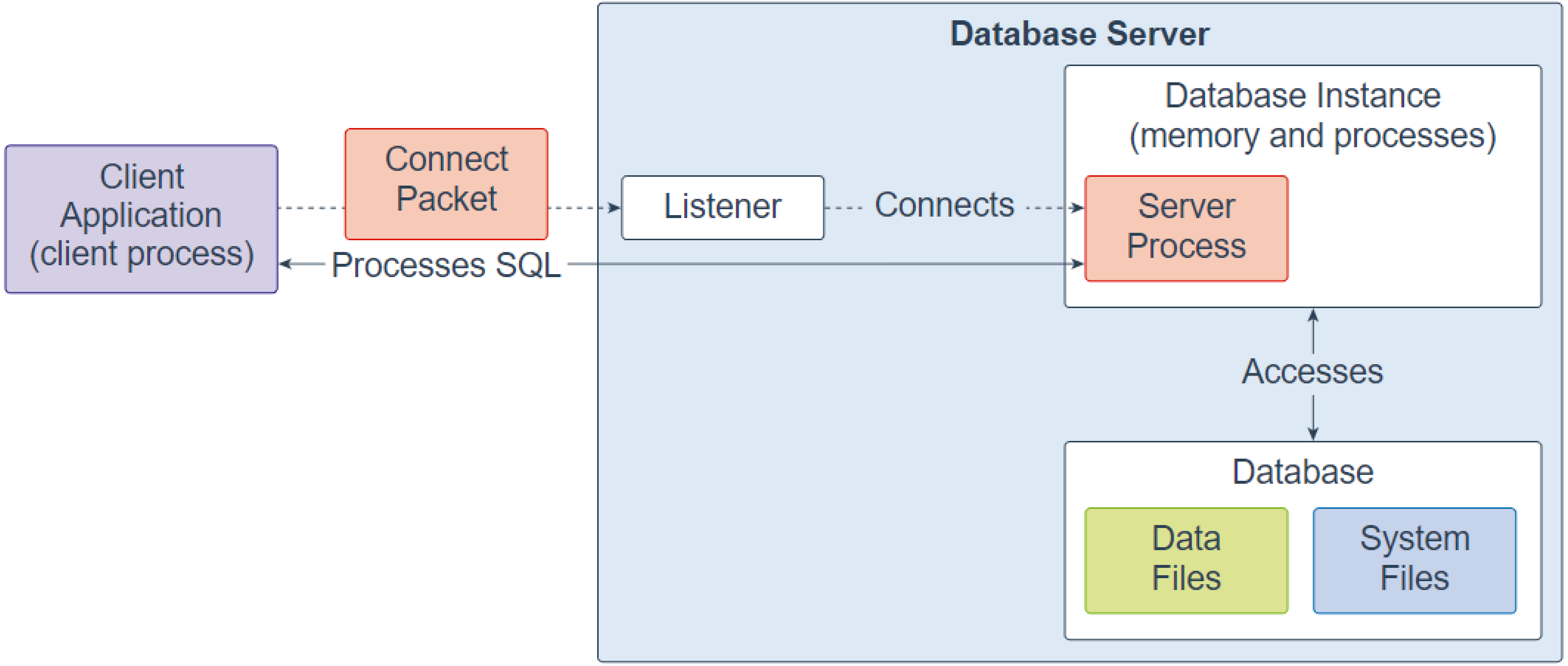
Initializing Extensions

oracle.bali.xml.addin.XMLDefaultRecognizerAddin

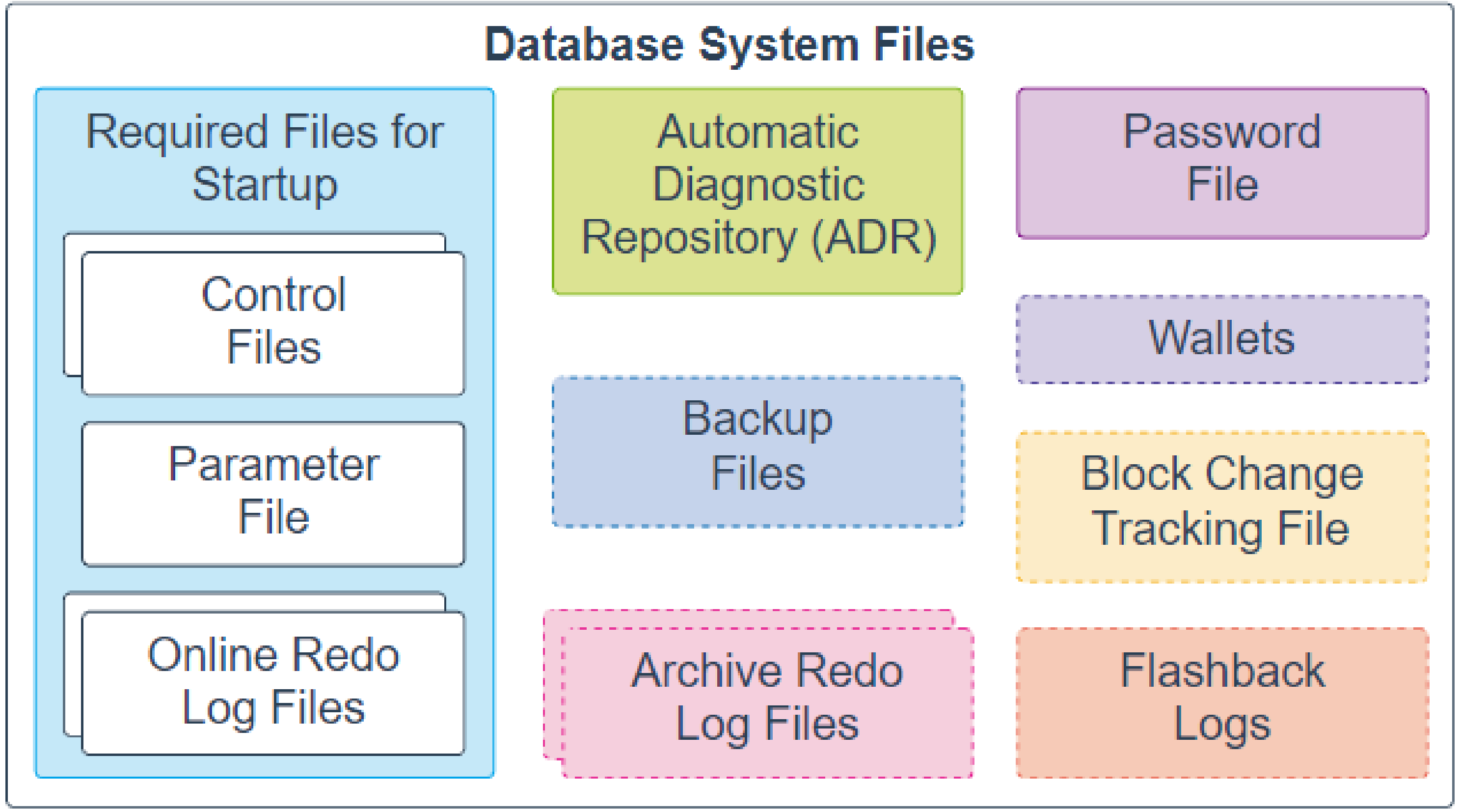
Version 22.2.1

Copyright © 2005, 2022, Oracle and/or its affiliates. All rights reserved.

Oracle Database Architecture



Database System Files



Install Oracle and Setup HR Database

Follow the

Lab Manual

One User – One Schema

A schema is a collection of database objects.
A schema is owned by a database user

Tablespace

- Oracle divides a database into one or more logical storage units called **tablespaces**.
- Oracle logically stores data in the tablespaces and physically stores data in datafiles associated with the corresponding tablespaces.

Sqlplus / as sysdba

We are now connected as the admin user of the database which has privileges to create more users.

Setting your user

- `SQL> CREATE USER (any user name with prefix c##) IDENTIFIED BY (any password);`
- e.g. create user c##myuser identified by 123; *you may create a user with your name or specific to the schema*
- `SQL> GRANT UNLIMITED TABLESPACE TO C##MYUSER;`
- `SQL> GRANT CONNECT, RESOURCE, DBA TO C##MYUSER;`

What is SQL

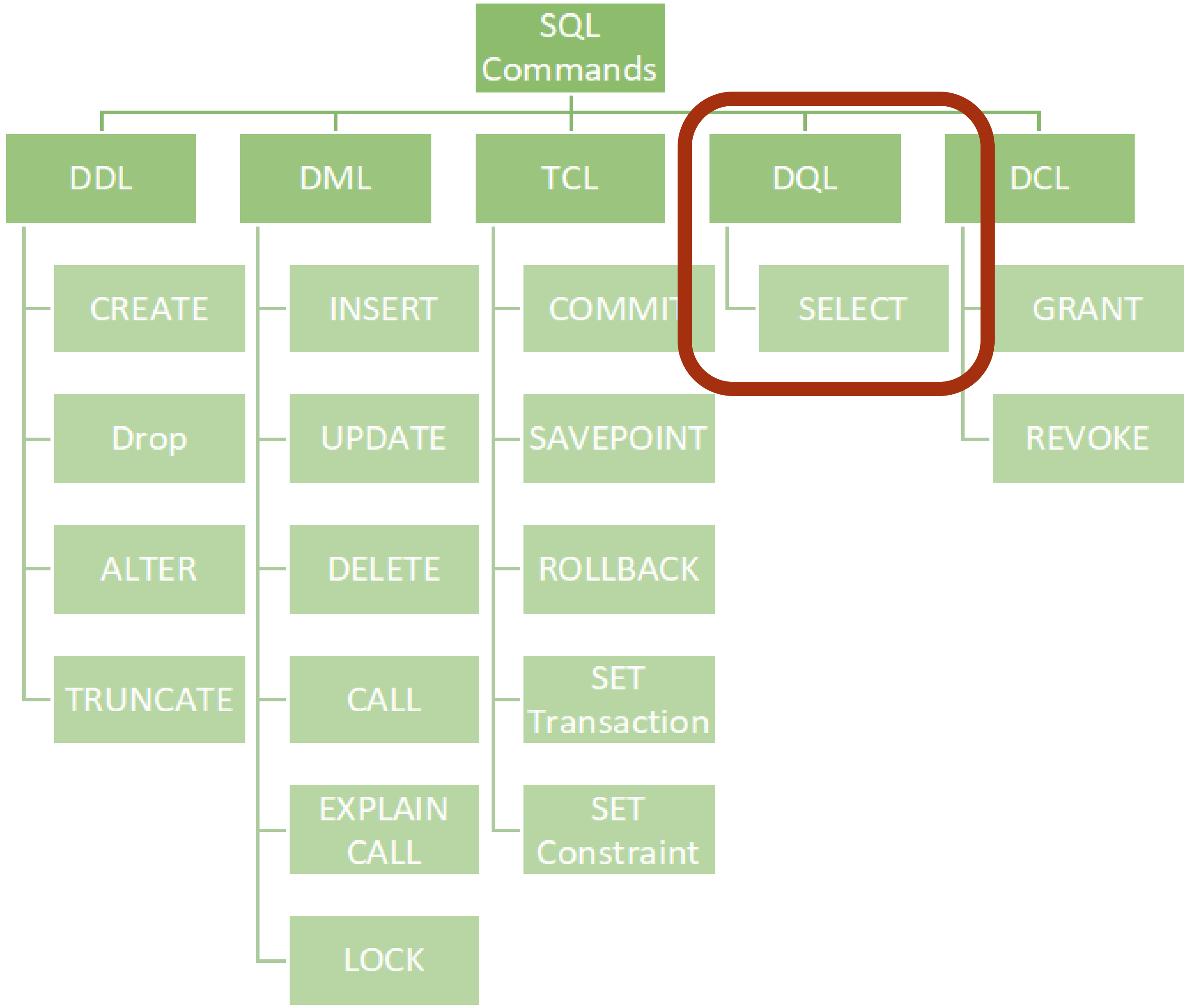
- Structured Query Language
- SQL is a standard language for querying and manipulating data

SQL

- Set-oriented and declarative – can retrieve/manipulate many records at a time (operates on sets of records instead of individual)
- SQL executed on CMD prompt or by programs in other languages.

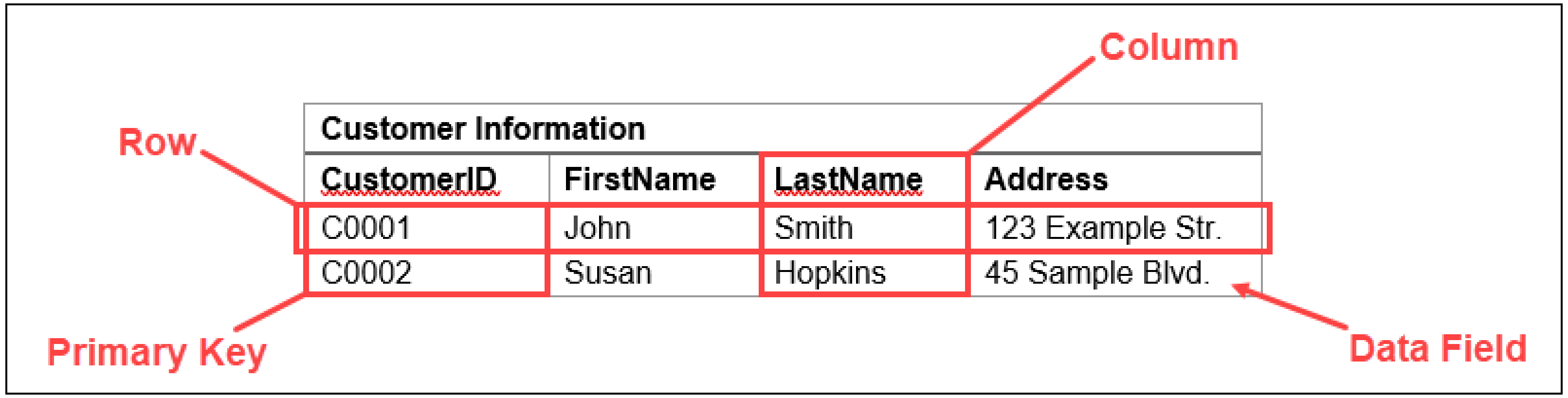
SQL Commands

- DDL – Data Definition Language
- DML – Data Manipulation Language
  - DQL – Data Query Language
- DCL – Data Control Language
- TCL – Transaction Control Language


SQL Commands
DDL
DML
TCL
DQL
DCL
CREATE
Drop
ALTER
TRUNCATE
INSERT
UPDATE
DELETE
CALL
EXPLAIN CALL
LOCK
COMMIT
SAVEPOINT
ROLLBACK
SET Transaction
SET Constraint
SELECT
GRANT
REVOKE

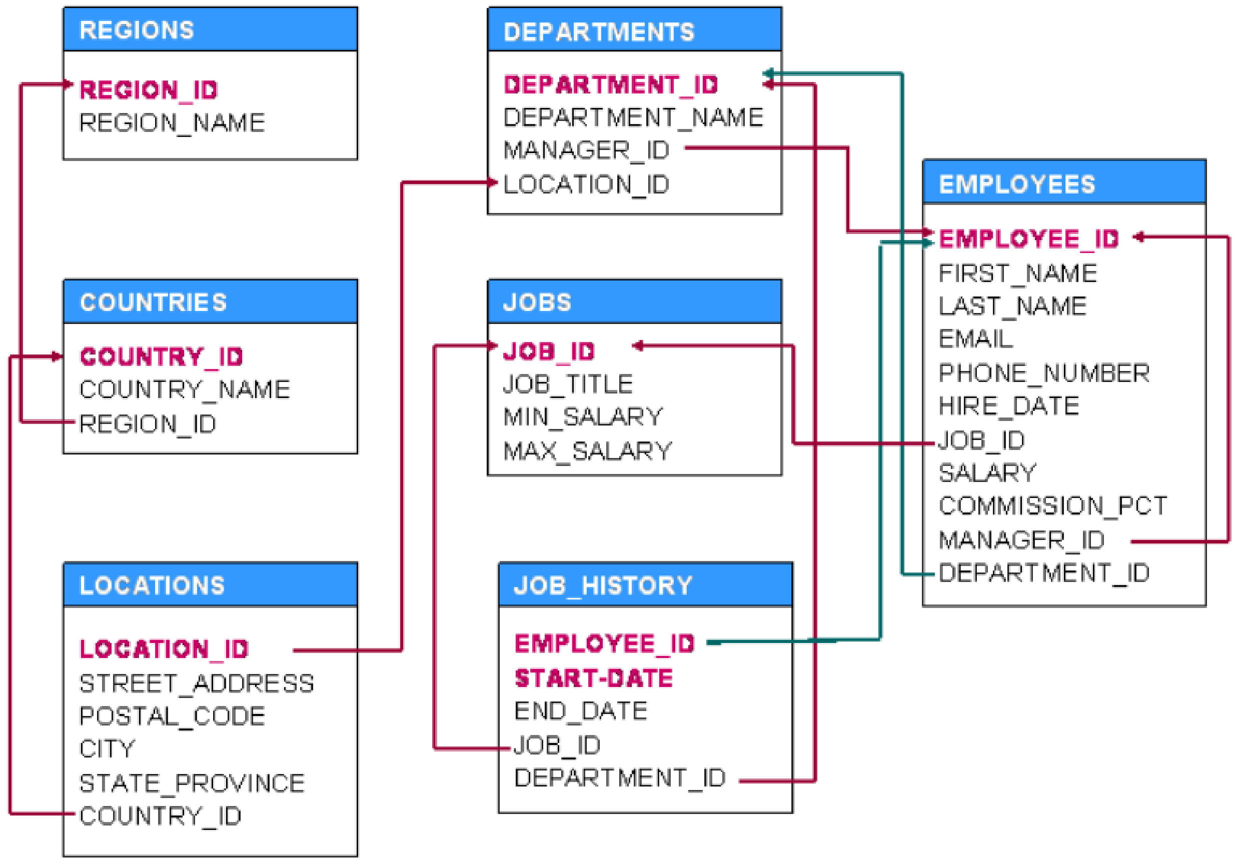
Tables

- A relation or table is a multiset of rows, with columns.



Customer Information			
CustomerID	FirstName	LastName	Address
C0001	John	Smith	123 Example Str.
C0002	Susan	Hopkins	45 Sample Blvd.

HR Schema


REGIONS
REGION_ID
REGION_NAME
COUNTRIES
COUNTRY_ID
COUNTRY_NAME
REGION_ID
LOCATIONS
LOCATION_ID
STREET_ADDRESS
POSTAL_CODE
CITY
STATE_PROVINCE
COUNTRY_ID
DEPARTMENTS
DEPARTMENT_ID
DEPARTMENT_NAME
MANAGER_ID
LOCATION_ID
JOBS
JOB_ID
JOB_TITLE
MIN_SALARY
MAX_SALARY
JOB_HISTORY
EMPLOYEE_ID
START-DATE
END_DATE
JOB_ID
DEPARTMENT_ID
EMPLOYEES
EMPLOYEE_ID
FIRST_NAME
LAST_NAME
EMAIL
PHONE_NUMBER
HIRE_DATE
JOB_ID
SALARY
COMMISSION_PCT
MANAGER_ID
DEPARTMENT_ID

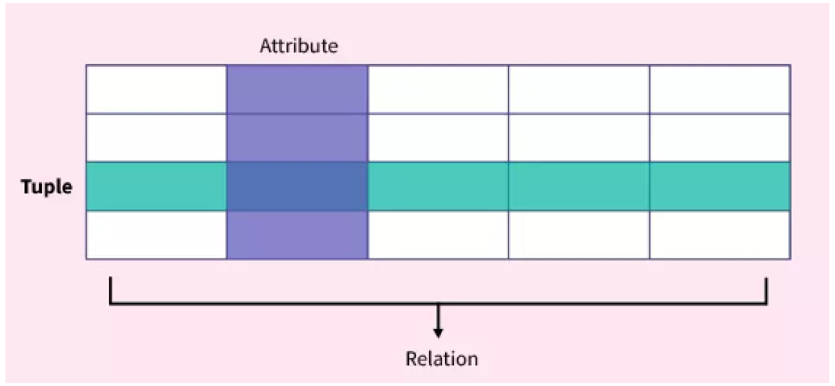
Syntax details

- SQL commands are **case insensitive -** SELECT = Select, Product = product
- Values are not, ‘Seattle’ not equal to ‘seattle’
- Use single quotes for constants : ‘abc’ - best practice (versus “abc” with mixed support)
- To say “don’t know the value”/ missing values we use **NULL** E.g., Student GPA in 1st quarter = NULL, not zero
- Free-form language – no special indentation is required but consistent formatting style is recommended for easy maintenance of SQL queries.

SELECT

SELECT c1, c2 FROM t;
Query data in columns c1, c2 from a table

SELECT * FROM t;
Query all rows and columns from a table



Projection is the operation of producing an output table with tuples that have a subset of their prior attributes

SELECT Example

StudentID	Name	HomeCity	PhoneNumber	Email
1	Alpha	Karachi	12345	alpha@iba.edu.pk
2	Bravo	Islamabad	23456	bravo@iba.edu.pk
3	Charlie	Karachi	34567	charlie@iba.edu.pk

- Table Name: Students
- Retrieve all information of all students
- SELECT * FROM Students;
- Retrieve all names of students
- SELECT Name FROM Students;

WHERE

Selection is the operation of filtering a relation's tuples on some condition

```
SELECT c1, c2 FROM t
WHERE condition;
```

Query data and filter rows with a condition

- Comparison Operators →

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. **Note:** In some versions of SQL this operator may be written as !=
BETWEEN	Between a certain range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

WHERE Example

StudentID	Name	HomeCity	PhoneNumber	Email
1	Alpha	Karachi	12345	alpha@iba.edu.pk
2	Bravo	Islamabad	23456	bravo@iba.edu.pk
3	Charlie	Karachi	34567	charlie@iba.edu.pk

- Retrieve all information of Alpha
- SELECT * FROM Students WHERE Name=‘Alpha’;
- Retrieve Phone Number and Email for Bravo
- SELECT PhoneNumber, Email FROM Students WHERE Name=‘Bravo’;

AND, OR, NOT

- Used in WHERE clause
- The AND operator : all the conditions are TRUE.
- The OR operator: if any of the conditions is TRUE.
- The NOT operator displays a record if the condition(s) is NOT TRUE.
- Take care of brackets when defining multiple conditions

AND/OR/NOT Example

StudentID	Name	HomeCity	PhoneNumber	Email
1	Alpha	Karachi	12345	alpha@iba.edu.pk
2	Bravo	Islamabad	23456	bravo@iba.edu.pk
3	Charlie	Karachi	34567	charlie@iba.edu.pk

- Find all students from Karachi or Islamabad
- SELECT * FROM Students WHERE HomeCity='Karachi' OR HomeCity='Islamabad';
- Find all students who are not from Karachi
- SELECT * FROM Students WHERE HomeCity<>'Karachi';
- SELECT * FROM Students WHERE NOT HomeCity='Karachi';

IN operator

- The `IN` operator allows you to specify multiple values in a `WHERE` clause.
- The `IN` operator is a shorthand for multiple `OR` conditions.

IN Example

StudentID	Name	HomeCity	PhoneNumber	Email
1	Alpha	Karachi	12345	alpha@iba.edu.pk
2	Bravo	Islamabad	23456	bravo@iba.edu.pk
3	Charlie	Karachi	34567	charlie@iba.edu.pk

- Find all students from Karachi or Islamabad
- SELECT * FROM Students WHERE HomeCity=‘Karachi’ OR HomeCity=‘Islamabad’;
- SELECT * FROM Students WHERE HomeCity IN (‘Karachi’ ,‘Islamabad’);
- Find all students not from Karachi or Islamabad
- SELECT * FROM Students WHERE HomeCity NOT IN (‘Karachi’ ,‘Islamabad’);

Distinct

```
SELECT DISTINCT c1 FROM t
WHERE condition;
```

Query distinct rows from a table

Distinct Example

StudentID	Name	HomeCity	PhoneNumber	Email
1	Alpha	Karachi	12345	alpha@iba.edu.pk
2	Bravo	Islamabad	23456	bravo@iba.edu.pk
3	Charlie	Karachi	34567	charlie@iba.edu.pk

- Find all unique city names
- SELECT DISTINCT HomeCity FROM Students;

HomeCity
Karachi
Islamabad

Aliases

- **Column Name**

```
SELECT column_name AS alias_name
FROM table_name;
```

- **Table Name**

```
SELECT column_name(s)
FROM table_name AS alias_name;
```

BETWEEN

```
SELECT c1, c2 FROM t
WHERE  c1 BETWEEN low AND high;
```

Query rows between two values

- Selects values within a given range. The values can be numbers, text, or dates.
- The BETWEEN operator is inclusive: begin and end values are included.
- Similar to querying using a combination of >= and <=

BETWEEN Example

StudentID	Name	Marks
1	Alpha	60
2	Bravo	78
3	Charlie	70

- Find all students who scored between 70 and 80
- SELECT * FROM Students
  WHERE Marks BETWEEN 70 AND 80;

StudentID	Name	Marks
2	Bravo	78
3	Charlie	70

LIKE (Wildcards)

```
SELECT c1, c2 FROM t1
WHERE c1 [NOT] LIKE pattern;
```

Query rows using pattern matching %, _

- The percent sign (%) represents zero, one, or multiple characters
- The underscore sign (_) represents one, single character

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
WHERE CustomerName LIKE 'a__%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

ORDER BY

```
SELECT c1, c2 FROM t
ORDER BY c1 ASC [DESC];
```

Sort the result set in ascending or descending order

- Default ascending order
- DESC – descending
- ASC - ascending

ORDER BY Example

StudentID	Name	HomeCity	PhoneNumber	Email
1	Alpha	Karachi	12345	alpha@iba.edu.pk
2	Bravo	Islamabad	23456	bravo@iba.edu.pk
3	Charlie	Karachi	34567	charlie@iba.edu.pk

- Sort the table by HomeCity
- SELECT * FROM Students
  ORDER BY HomeCity;
- Sort the table by column number 1
- SELECT * FROM Students
  ORDER BY 1;

IS NULL

- NULL indicates data is unknown, inapplicable or does not exist i.e. refers to missing data

marks = NULL	☒	***marks IS NULL***	☑
marks <> NULL	☒	***marks IS NOT NULL***	☑