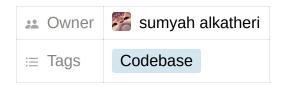


## notes





## introduction

introduction

#### Data

Database

Database management system websites can have two types:

oracle features:

DAY 2:

1.ER model

One-to-One (1:1) Relationship:

2. Schema

#### 3.Class diagram

Database design phases,

#### Relationships between the tables or objects:

Classification of sql commands:

Data types in the sql database

#### **SQL Command types**

DDL

In creating a constraint there is two ways: inLine and out of line:

#### the commands:

Case1: if you have created already the table, and you want to add a new column

Case 2 : if you have created already the table, and you added a constraint to a column and you want to change that constraint to a different one

CASE 3: if you want to add a constraint not null or unique to a column that already exists in a table

CASE 4 :IF YOU ARE GONING TO DO AN OUTLINE CONSTRAINT AFTER WRITING ALL THE COLUMNS

#### TRUNCATE COMMAND

Rename table

#### TYPES OF CONSTRAINTS:

check constraint

FOREIGN KEY CONSTRAINT TYPOES:

TCL—>Transaction control language

CH5-12



Day 1:

#### **ERRORS**

## **Data**

can bs stored in either software or hardware

Hardware means physical devices such as hard drive and software —> cloud

#### **Database**

DBMS stands for <u>Database Management system</u> which is software that allows users to build, create and manipulate the database, allowing them to process, analyze and store data easily.

DBMS provides a tool or interface to perform different operations such as storing data, creating and updating it, and creating tables in the database.

## **Database management system**

are software applications that allow users to manage and manipulate data in databases. it has two types:

#### ▼ Relational Database Management System (RDBMS):

It stores data in tables, rows are called records and the column are called attributes and the table is called an entity.

#### **▼** Not relational (NoSQL Database Management System:

NoSQL (Not Only SQL) databases are designed to handle unstructured or semistructured <u>data.it</u> uses arrays and dictionaries.

EX: **Document Databases:** Store data in **JSON-like** documents, allowing flexibility in data structure. Examples include MongoDB and Couchbase.

## websites can have two types:

- 1. **Dynamic**: when making changes to the website, we are not required to change the code
- 2. **Static**: to make any changes or updates, changes to the code are required
- 3. combination

## oracle features:

- 1. oracle is a cross-platform: can run in any operating system IOS, windows, ubuntu
- 2. oracle also has a network stack which means it is compitable in any application with diffrent os systems

3. DBMS ensures data integrity which means data is free from errors and bugs.

#### 4. ACID:

These are the four key properties that ensure the reliability and integrity of transactions in a database management system (DBMS).

#### • A —>Atomicity

If any part of the transaction fails, the entire transaction is rolled back, and the database remains unchanged.(indivisible units of work)

For example: you are purchasing a product from an online website and you added your product to the shopping list, and after adding the first product you thought of adding another item to the list however this time you get an internet connection interruption and the transaction of adding the second item doesn't happen \and stays in the inventory so no changes happened in this case

due to this error you cant continue into this transaction which is purchasing the products .

#### • C —> Consistency

This means that the database must satisfy a set of predefined integrity constraints before and after the transaction. simply it means that the nothing cant be added to the database if it doesn't comply to the rules . For example: if you have table for students info ,it has a column for the birthdate and you have an if statement of only accepting students that are 15 years and older . then the DBMS wont accept students younger than 15 years old. this makes the database consistent.

if years≥15:
The DBMS will accept the student's record.
else:
The DBMS will reject the student's record.

- I —> Isolation
- . The isolation property ensures that each transaction appears to be executed in isolation from other transactions.
  - D —> Durability

This means that once a transaction is committed, its changes must persist even if there is a system failure. The durability property ensures that the database will not lose data due to system failures or crashes.

**Validation** of the data can be done from both the customer and the employer perspectives in the context of the Software Requirements Specification (SRS) document.

## **DAY 2:**

## 1.ER model

represents a design of a database that can later be implemented as a database and **displays the relationships of entities** stored in a database.

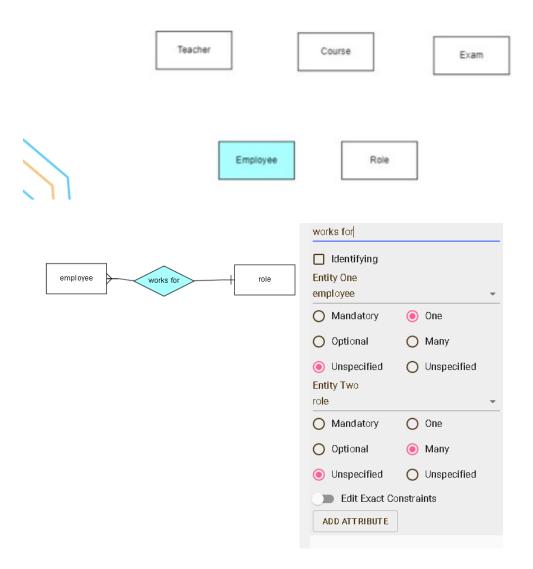
## One-to-One (1:1) Relationship:

- A one-to-one relationship means that each instance of one entity is associated with only one instance of another entity, and vice versa.
- example: every customer has his own id card. every instance from the customer entity belongs to one instance in the id card entity.

## **Example:**

In a LMS System, there are many courses teach by a many teachers. Each course has a name, status, date from and date to. Each course has an exam. In this system, there are many employees work in different positions.

1. Firstly, create the LMS entities.



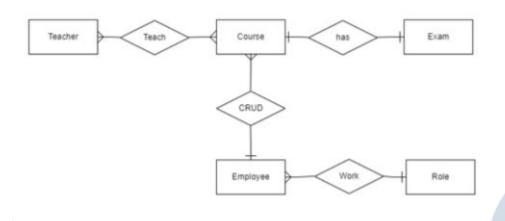
## the way we read this is like the following:

1)the first part from left, each employee works for one role (so it is a one to one from the role side)

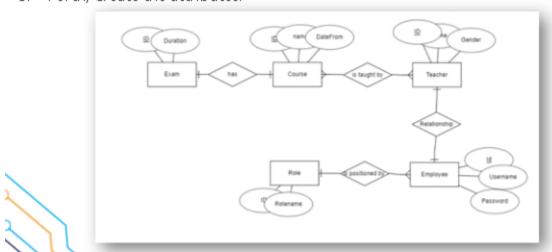
2)the second part to the right, each role is assigned by many employee(so many from the employees side)



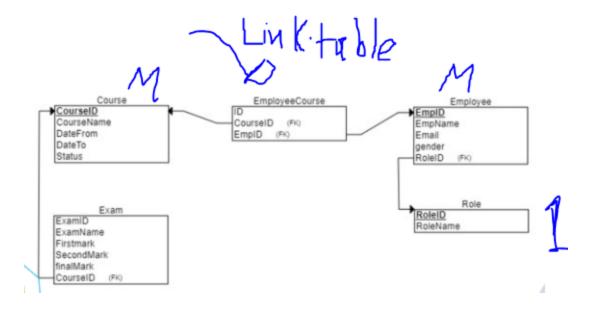
note: An Entity-Relationship Diagram (ERD) is typically read from **top to bottom** and from **left to right.** In an ERD.



3. Forth, Create the attributes.

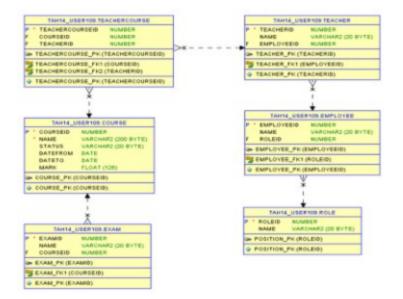


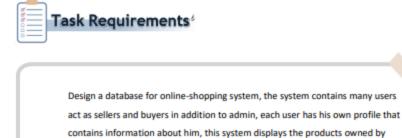
## 2. Schema



## 3.Class diagram

A class diagram is used in modeling and designing software to describe classes and relationships.





1)user and role (1 to many) foreign key in users

for it.

2)user have products(1 to many) fk in products

3)user does payment(1 to many) fk in payments

4)product and cateogry(1 to many) fk in prosuct

#### 1. first we create the tables

```
CREATE TABLE UserTable (
user_id NUMBER(10) generated by default as identity primary key ,
user_firstname VARCHAR2(100),
user_lastname VARCHAR2(100)
);
```

sellers based on their categories, the sellers can use this service when they pay

we add the foreign key profile\_id , by using the ddl command ALTER—>ADD CONSTRAINT

ALTER TABLE UserTable #THE TABLE NAME

ADD CONSTRAINT fk\_student\_profile\_id #CREATE A NAME FOR THE FOREIGN KEY

FOREIGN KEY (profile\_ID) REFERENCES profile(profile\_id)# CREATE THE COLUMN NAME

#OF THE FOREIGN KEY profile\_ID IN THE USERTABLE AND THEN REFENCE TO THE

#PROFILE TABLE PRIMARY KEY

## Database design phases,

- Requirement analysis.
- ER Diagram.
- Schema Diagram.
- Class Diagram.

## Relationships between the tables or objects:

- 1. Many-to-Many: Course and Teacher.
- 2. One-to-One: Course and Exam.
- 3. One-to-Many: Role and Employee.



A **foreign key** is a concept in relational databases that establishes a link between two tables based on the values of a column or columns. the foreign key references back to the primary key which is found in the base table.

one-to -one	one -to many	many to many
-each instance in the first table relates to one instance in the other table.	-each instances relates to many instances in other table. but not vice versa	-each instances relates to many instances in other table and vice versa
EX: each customer from the customer table has his own id from the id table.	EX: employee and position, each employee has one position, and each position has many employees assigned to it	EX: courses and doctors, each course is taught by many doctors , and each doctor teaches many courses
-the foreign key can be assigned in either of the two tables	-the foreign key has to be assigned on the many side of the relationship, for the above example the foreign key will be with the employees table.	-a third table is created to put the foreign key in it. it will have courses_id_fk and doctors_id_fk relating to the primary keys ids in the doctors and courses table

## **Classification of sql commands:**

**Ddl** command is used for the structure of the database **Dml** is the management and editing , adding the data of the database

**Tcl** is happens for autosaving and the commit when there are any changes. **dcl** admin giving privalages and permissions to other users **dql** queries such as select, basically retrieving the data only

The tables or objects in the database of sql are converted to a class when we connect it to the back-end, the mvc or mistro(the backend servers application has a documentation that explain the process of converting the database into the backend.

The backend servers database datatypes has to be same as the database in the local sql server

## Data types in the sql database

1. The **number()** command in sql database is equivalent to double in the backend server.

The sql database data types should be the same as the datatypes in the backend to

Relational operations

1. are the greater and the les than

Th logical operators are the not ,and , or

Important :search more about the process of connecting

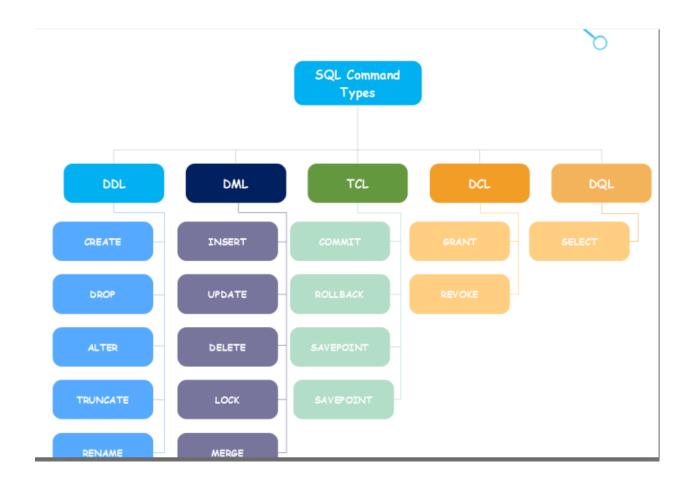
Database with backend.

**Varchar vs varchar2** it is equivalent to a string in the backend server.

Sql database doesn't have Boolean datatype, it uses zeros and ones to represent them.

2. **Data()** is equivalent to datetime method in the c # function.

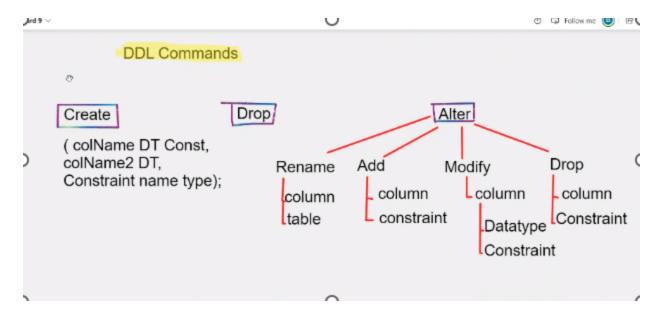
## **SQL Command types**

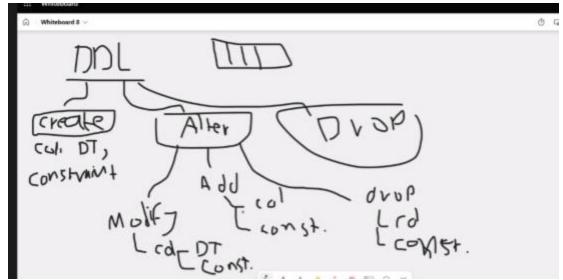


## **DDL**

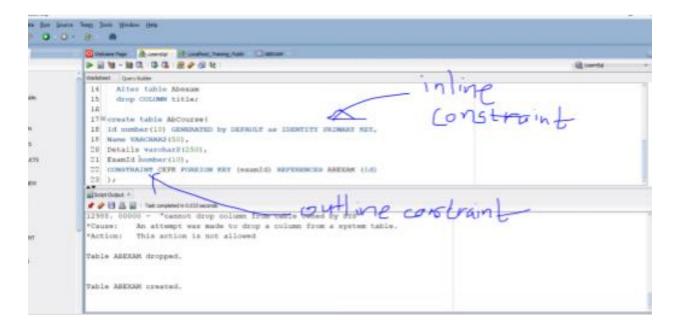
## Overview of Data Definition Language (DDL)

Data Definition Language (DDL) is used to create and updates the object structure in a database using specific syntax and predefined commands. These database objects include sequences, tables, locations, schemas aliases, and indexes.





In creating a constraint there is two ways: inLine and out of line:



- -INLINE CONSTRAINT : it is when we write the constraint directly beside the column
- -OUTLINE CONSTRAINT : it is when you write the constraint after creating all columns at the end .
- 1)alter modifying is only done for the column , not the constraint DDL is applied in all objects, such as the tables, packages, procedures.

## the commands:

## Case1: if you have created already the table, and you want to add a new column

ALTER TABLE table\_name ADD column\_name data\_type;



create table AbCourse( Id number(10) GENERATED by DEFAULT as IDENTITY PRIMARY KEY,

we use the phrase "GENERATED by DEFAULT as IDENTITY PRIMARY KEY" whenever ther

# Case 2: if you have created already the table, and you added a constraint to a column and you want to change that constraint to a different one

step1: You'll need to drop the existing constraint that you want to change

step 2: ADD A NEW CONSTRANT USING ALTER, ADD

#### CREATE TABLE YourTable (

examld INT.

-- other columns

CONSTRAINT CEFK FOREIGN KEY (examld) REFERENCES ABEXAM (id) );

in this example we are adding the constraint (outline constraint) after the colums

#### ALTER TABLE YourTable

ADD CONSTRAINT CEFK FOREIGN KEY (examid) REFERENCES ABEXAM (id);

in this example after we created the table already , we wanted to add a constraint to one of the columns

## CASE 3: if you want to add a constraint not null or unique to a column that already exists in a table

alter TABLE tablename

MODIFY column name (datatpe()of that column) not null;

## CASE 4 :IF YOU ARE GONING TO DO AN OUTLINE CONSTRAINT AFTER WRITING ALL THE COLUMNS

FOREIGN Key (foreign\_columnname) references primarykeytablename(primarykeycolumnname)

```
CREATE TABLE employee
( employee_id number(10) NOT NULL,
  employee_name varchar2(50) NOT NULL,
  email VARCHAR2(50) NOT NULL UNIQUE,
  salary float,
  address varchar2(50) DEFAULT 'Amman',
  age number(5),
  check (age>=18),
  PRIMARY KEY (employee_id),
  P_id number(10),
  FOREIGN KEY (P_id) REFERENCES position(position_id) );
```

#### **DESCRIBE COMMAND**

Describe tablename

### TRUNCATE COMMAND

**TRUNCATE** to remove all records from a table, including all spaces allocated for the records are removed.

TRUNCATE TABLE tablename

#### Rename table

alter table tablename rename to newtablename;

#### **ENABLE & DISABLE CONSTRAINT**

The disable constraint —> disables the function of a constraint temporarily

The enable constraint —> activates the disabled constraint back

#### **Enable and disable Constraint:**

Alter table table\_name **enable** | **disable** constraint constraintName

## **TYPES OF CONSTRAINTS:**

## **Constraints**

The following are commonly used constraints available in SQL:

- NOT NULL: Ensures that a column cannot have a NULL value.
- DEFAULT: Provides a column default value when none is specified.
- UNIQUE: Ensures that all values in a column are different.

#### check constraint

If you define a **CHECK** constraint on a column it will allow only certain values for this column.



can be applied for any data type

```
CREATE TABLE Persons (
ID int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Age int,
CHECK (Age>=18)
);
alter table abcourse
```

add CONSTRAINT checkCn CHECK (name in('DB','CS','OOP'));



While it is possible for **foreign key values** to temporarily not match primary key values, it's important to maintain data integrity and resolve such inconsistencies promptly to ensure the database's reliability and accuracy.

### **FOREIGN KEY CONSTRAINT TYPOES:**

#### 1. ON-DELETE CASCADE

#### **Defining Foreign Key Options**



2. ON DELETE SET NULL clause sets all the rows that are defined as a foreign key in the child table to Null where the row in the parent table is deleted.

Example: FOREIGN KEY (P\_id) REFERENCES position(position\_id) ON DELETE
SET NULL.

#### 2.ON -DELETE NULL

## TCL—>Transaction control language

Transaction Control Language (TCL) is used to manage all the transactions in the database. These are used to manage the changes made by Data manipulation language (DML) commands. It also allows commands and statements to be grouped together into logical transactions.

TCL is to mange the dml commands

dml commands : select , update ,INSERT, DELETE, merge

### **Transaction Control Language (TCL) Commands**

- COMMIT: is used to save any transaction into the database.
- ROLLBACK: restores the database to the latest committed state. Also, It is also used in the savepoint command to move to a savepoint in a transaction.
- SAVEPOINT: is used to tentatively save a transaction, so the user can roll back to the save point whenever necessary.

(ddl)**Drop** —>deletes the whole structure of the table

(ddl)**Tranciate** —> ony deletes the rows , the structure stays the same

(dml) **delete** —> it deletes all rows , keeping the structure the same.

## dml command syntax:

```
UPDATE table_name
SET column1 = new_value1, column2 = new_value2, ...
WHERE condition;
```

## CH5-12

CHP 5-12