

جامعة
جامعة

MICROSOFT SQL SERVER

MICROSOFT SQL SERVER IS A RELATIONAL DATABASE MANAGEMENT SYSTEM. AS A DATABASE SERVER THAT STORES AND RETRIEVES DATA.

- * يخزن ويعتبر البيانات

- * MAKE YOUR OWN DOCUMENTATION.
- * VERY IMPORTANT. TAKE PHOTO TO PAGES.

* ERD STANDS FOR ENTITY RELATIONSHIP DIAGRAM.

* ENTITY RELATIONSHIP MODEL.

• يمثل العلاقات والبيانات.

* AN ERD VISUALIZES THE RELATIONSHIPS BETWEEN ENTITIES LIKE PEOPLE, THINGS OR CONCEPTS IN A DATABASE.

* DATABASE DESIGNER. PREPARE A ER-D DIAGRAM FOR THIS COMPANY ACCORDING TO THE FOLLOWING DESCRIPTION.

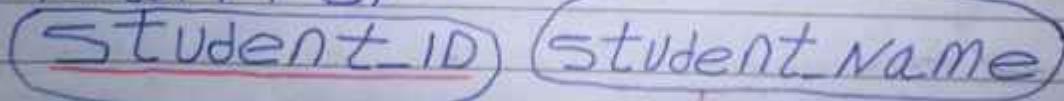
* THE COMPANY HAS A NUMBER OF EMPLOYEES EACH EMPLOYEE HAS SSN, BIRTH DATE, GENDER AND NAME WHICH IS REPRESENTED AS FNAME, LNAME.

* THE COMPANY HAS A SET OF DEPARTMENTS. EACH DEPARTMENT HAS A SET OF ATTRIBUTES DNAME, DNUMBER (UNIQUE) AND LOCATIONS.

* EMPLOYEES WORK IN SEVERAL PROJECTS. EACH PROJECT HAS PNAME, PNUMBER (IS AN IDENTIFIER), LOCATION AND CITY.

- * Each employee may have a set of dependent; each dependent has dependent name (unique), gender and birth date.
Note: if the employee left the company no needs to store his dependents info.
- * For each department, there is always one employee assigned to manage that department and each manager has a hiring date. $1 \leftrightarrow M$
- * Department may have employees but employee must work on only one department
- * Each department may have a set of projects and each project must be assigned to one department. $1 \leftrightarrow M$
- * Employees work in several projects and each project has several employees and each employee has $M \leftrightarrow M$ a number of working hours in each project
- * Each employee has a supervisor and supervisor has many employee. $1 \leftrightarrow M$

Primary Key:



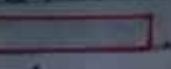
The key is underlined
is Primary Key

STUDENT

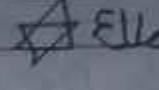
BACK to (System Analyst) ER Description.

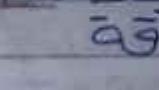
ER Diagram

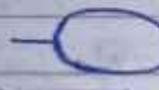
* SUMMARY OF NOTATION FOR ER DIAGRAMS

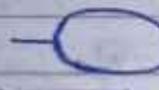
1. ENTITY 

2. WEAK ENTITY 

3. RELATIONSHIP   

4. IDENTIFYING RELATIONSHIP   

5. ATTRIBUTE    SIMPLE.

  circle

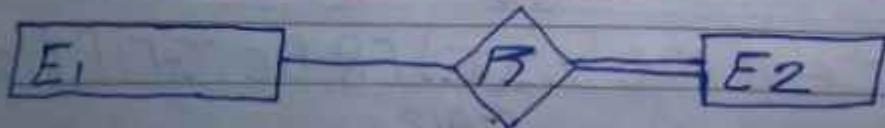
6. KEY ATTRIBUTE 

7. MULTIVALUED ATTRIBUTE 

8. COMPOSITE ATTRIBUTE 

9. DERIVED ATTRIBUTE 

10. TOTAL PARTICIPATION OF E2 IN R.



11. CARDINALITY RATIO 1:N FOR E1:E2 IN R.



Participation Case Study

1. May, Zero or More, OPTIONAL = Partial.
 2. One or More, MUST, MANDATORY = Total.
- * the Relationship cardinality 1..M.

* DIFFERENCE TYPES OF KEYS :-

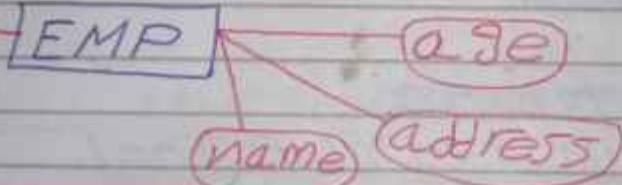
1. Candidate Key.
2. Primary Key.
3. Foreign Key.
4. Composite Key.
5. Partial Key.
6. Alternate Key.
7. Super Key.



instructor



ID



SSN

BD

Gender

FN

LN

EMP

Name

age

address

DRUM

DRUM

DName

Loc

Supervisor

M

work

1

Dept

1

Manage

hiredate

NOS

Loc

Dependent

DMme

ED

gender

hours

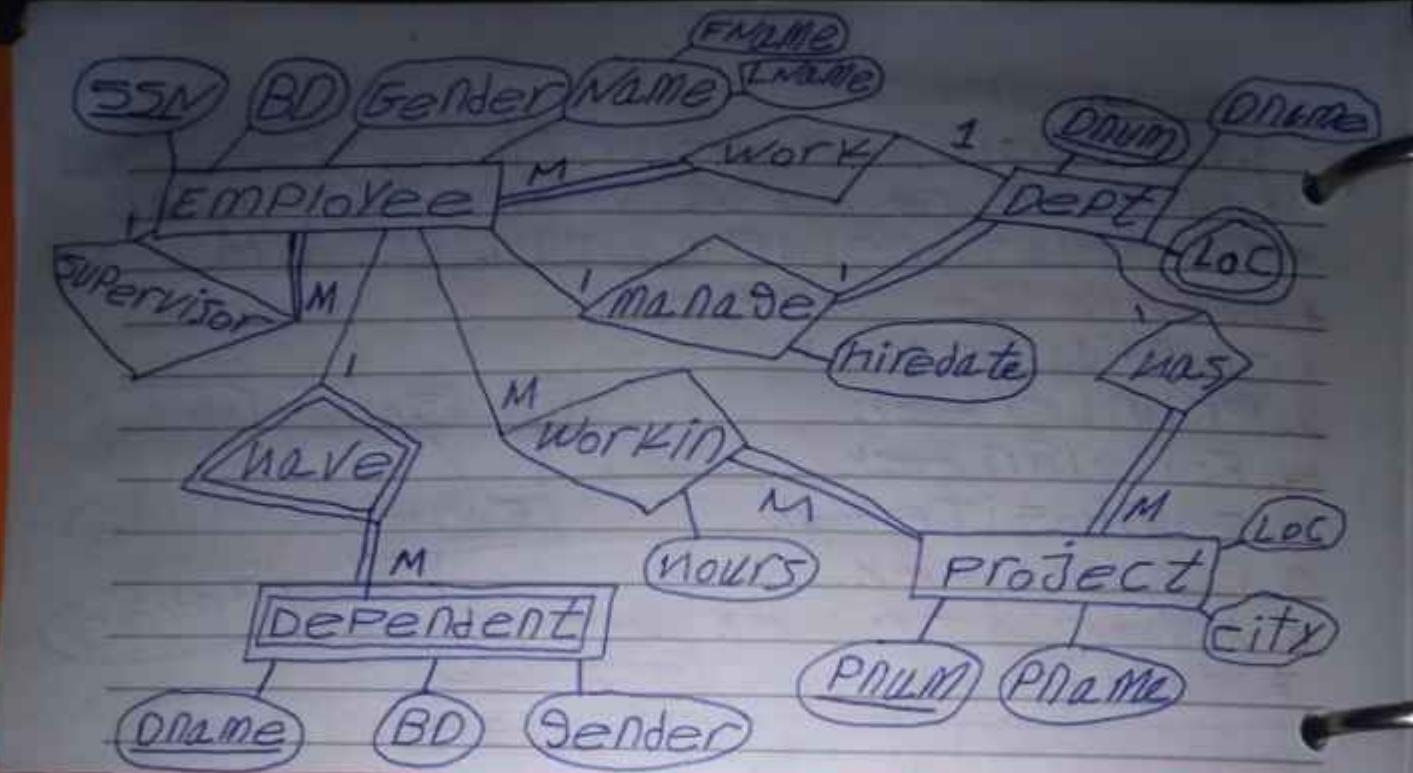
Project

Loc

PNUM

PNAME

city



Become a Data Analyst:

1. Maths & Statistics.
2. EXCEL BASICS.
3. PYTHON.
4. SQL & Database.
5. Data visualization Tool (PowerBI / TABLEAU).

Data Base Life Cycle:-

1. ANALYSIS → SYSTEM ANALYST. Req'DOC.
2. DB DESIGN → DB DESIGNER.
3. DB MAPPING → DB DESIGNER.
4. DB SCHEMATIC (TABLES & RELATIONSHIPS).
5. DB IMPLEMENTATION → DB DEVELOPER.
6. SQL (PHYSICAL DB). SQL QUERY.
7. APPLICATION → APPLICATION PROGRAMMER.
8. GUI -- INTERFACE. WEB-DESKTOP-MOBILE.
9. CLIENT → END USER.

ERD Entity relationship model

1. ENTITY - Strong Entity PK.

Weak Entity Partial Key.

Primary Key ID

Partial Key

Attributes → Simple

Composite.

CMCC

Multivalued Attributes

Computed. SCMCC.

Complex.

Relationship

Degree Unary ✓

Binary ✓

Ternary ✓

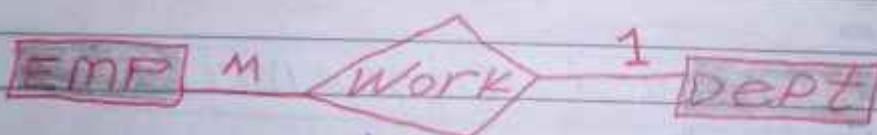
Cardinality 1 1. ✓

1 M. ✓

MM. ✓

Participation (Total-Partial).

DB MAPPING



Y int Foreign Key

Eid	Ename	Did
1	Ahmed	10
2	Ali	300
3	Eman	10
		NULL

X int

Did	Dname
10	SD
20	IS
30	CS



FK PK

Relational Database Definitions

1. Table or Entity: - a collection of records.
2. Attribute or Column: - characteristic of an entity.
3. Row: specific characteristics of one entity.
4. Database: a collection of tables.

MAPPING -> DataBase Tables

CUSTOMER

customerID	customerName	Address
city	state	ZIP

Primary Key:

ORDER

orderID	orderDate	customerID
---------	-----------	------------

Foreign Key

orderLine

orderID	Product_ID	quantity
---------	------------	----------

Composite Primary Key:

ID

NAME

Address

SKILL

Multivalued Attribute

EMPLOYEE

EMPLOYEE

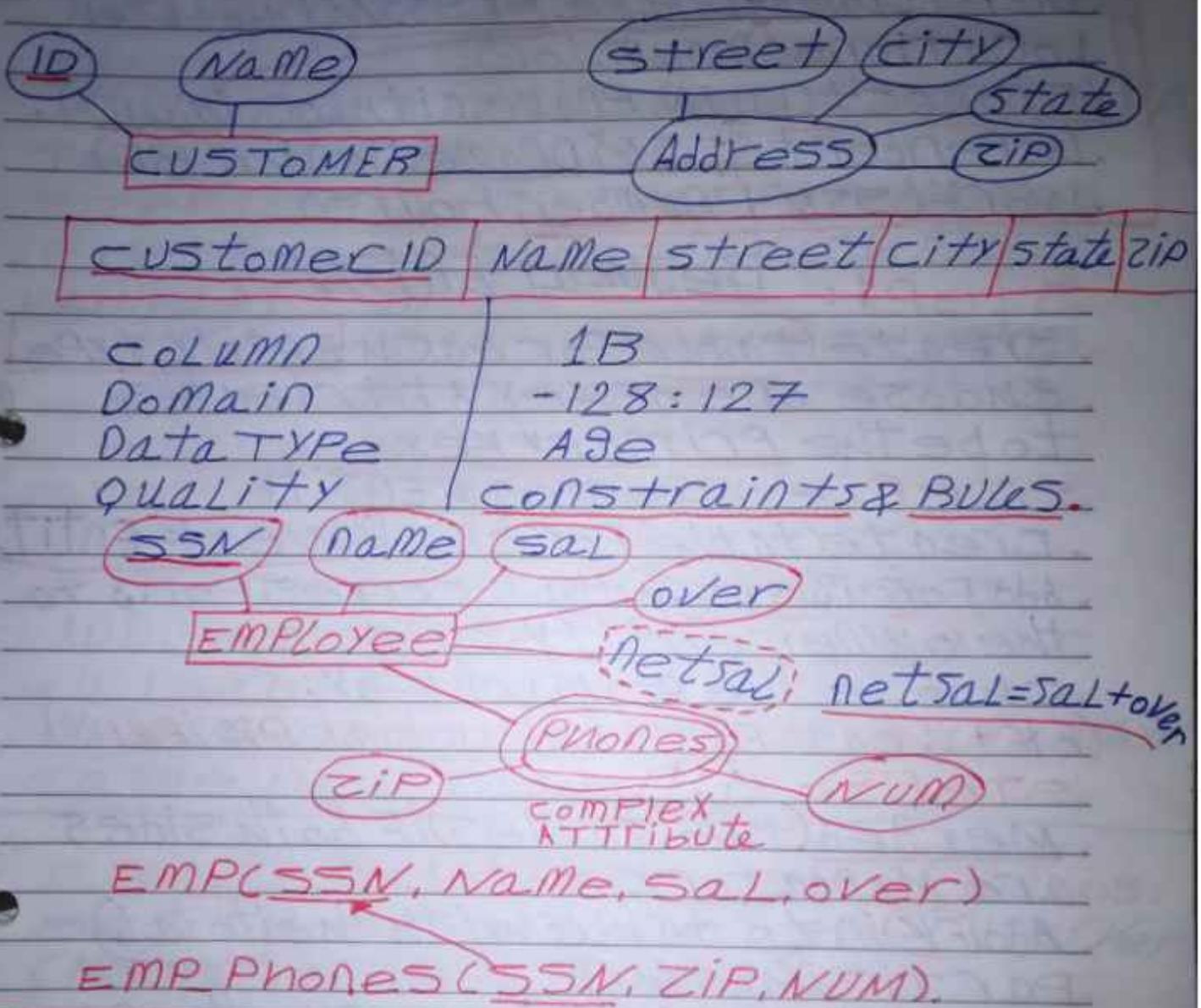
Employee_ID	Name	Address
-------------	------	---------

EMPLOYEE_SKILL

Employee_ID	skill
-------------	-------

New Table

composite attribute.



MAPPING

- Employee(SSN, BD, Gender, FN, LN, DNo, SuperId)
- Dept(Dnum, DName, MgrSSN, hiredate)
- Locations(Dnum, Loc)
- Project(Pnum, Pname, City, Loc, Dnum)
- Dependent(SSN, DName, Gender, BD)
- Work(SSN, Pnum, hours)

Step 1 :- Regular Entity

- create table for each Entity type
- choose one of key attributes to be the Primary key.

Step 2 :- weak Entity

- create table for each weak Entity
- Add Foreign Key that corresponds to the owner entity type.
- choose the primary key:

(FK + weak Entity Partial PK if any)

Step 3 :- 1:1

- Merged (two tables) if both sides are mandatory. total.

- Add FK into table with the total participation.

- Create third table if both sides are optional. Partial.

Step 4 :- 1:N

- ADD FK to N-side table.

- Add any simple attribute relationship as column to N-side table.

M:M.
Step 5: M:N

- Create a new Third table.
- Add FKs to new table for both Parent tables.
- ADD simple attributes of relationship to the new table if any.

Step 6: Multi-valued attributes.

- Create New table for each multi-valued attribute.
- Table will include two columns, one for multi-valued attribute + FK column.

Step 7: N Relationship.

- If $n > 2$ then:
- Create a new Third table.
- ADD FKs to the New table for all Parent tables.
- ADD simple attributes of relationship to the New table.

Create Relationships and Tables:

- Search → Services → SQL Server (M550LServer). Running.
- SQL Server → Server name: 192.168.10.120 → Connect.
- The American National Standards Institute SQL.
- We will use Microsoft SQL Server

MAPPING

- ~~Employee(SSN, BD, Gender, FN, LN, DNo, Superid)~~
- ~~Dept(Dnum, Dname, MgrSSN, hiredate)~~
- ~~Locations(Dnum, Loc)~~
- ~~Project(Pnum, Pname, city, Loc, Dnum)~~
- ~~Dependent(SSN, Dname, Gender, BD)~~
- ~~Work(SSN, Pnum, hours)~~

REGULAR ENTITY

Step 1:-

- create table for each entity type
- choose one of key attributes to be the Primary Key.

Step 2:-

WEAK ENTITY

- create table for each weak entity
- Add foreign key that correspond to the owner entity type.
- choose the primary key:

(FK + weak entity Partial PK if any)

Step 3:- 1:1

- Merged (two tables) if both sides are Mandatory. Total.
- Add FK into table with the Total participation.
- Create third table if both sides are optional. Partial.

Step 4:- 1:N

- ADD FK to N-side table.
- Add any simple attribute relationship as column to N-side table.

M:M.

Step 5:- M:N

- Create a new Third table.
- Add FKs to new table for both Parent tables.
- ADD SIMPLE ATTRIBUTES OF RELATIONSHIP to the new table if any.

Step 6: Multi-valued Attributes.

- Create New table for each MULTI-VALUED ATTRIBUTE.
- Table will include two columns, one for MULTI-VALUED ATTRIBUTE + FK COLUMN.

Step 7: N Relationship.

- IF $n > 2$ then:
- Create a new Third table.
- ADD FKs to the New table for all Parent tables.
- ADD SIMPLE ATTRIBUTES OF RELATIONSHIP to the New table.

Create Relationships and Tables:

- Search → Services → SQL Server (M550LServer). Running.
- SQL Server → Server Name: 192.168.10.120 → Connect.
- The American National Standards Institute SQL.
- We will use Microsoft SQL Server.

American National Standards Institute ANSI was original developer.

ANSI SQL:

1. Open Source MySQL.
2. IBM - PL-SQL.
3. Oracle PL-SQL.
4. Microsoft Transact-SQL.

MICROSOFT TRANSACT-SQL:

1. DDL. Data Definition Language.
Metadata & Structure.
 1. Create Table. ✓
 2. Create View. ✓
 3. Create Function. ✓
 4. Alter. ✓
 5. Drop. ✓
 6. Select Into. ✓

2. DML. Data Manipulation Language.

DATA.

1. Insert. ✓
2. Update. ✓
3. Delete. ✓
4. Merge. ✓

3. DCL. Data Control Language.

SECURITY & PERMISSIONS.

1. Grant. G. ✓
2. Deny. D. ✓
3. Revoke. R. ✓

4 DQL Data query language

- DISPLAY
- 1. Select. SELECT. 1
- 2. +
- 3. AGG FUN. AGG FUN. 2
- 4. GROUPING. Grouping. 3
- 5. UNION. UNION. 4
- 6. JoINS. JoINS. 5
- 7. SUBQUERIES. Subqueries. 6

5. TCI Transactional Control Language

- EXECUTION
- 1. begin transaction. BEGIN.
- 2. COMMIT. COMMIT.
- 3. ROLLBACK. ROLLBACK.

• DATABASES (right click) -->

New Database --> Database Name: SD.

• TABLES (right click) --> New --> Table

--> INPUT (ColumnName, DataType, ALLOWNULLS)

. then set primary key constraint -->
Save TableName. 19.

* choose name.

* Enter a name for the table:

* OK.

• the Relationships go to Database

Diagrams --> New Database Diagrams

-->> ADD Table -->> Select All Tables
ADD.

* To make relationship between tables

* Then save diagram.

Good Developer

- read Error & solve it.
- * Right Click Edit to ~~Top 200 Rows~~.
- * Input Data into Tables.

Code - SQL

USE SD;

Select *

From Employee;

ColumnName - DataType - Constraints:

Create Table emp

eid int Primary Key,

ename varchar(20) Not Null,

cage int,

cadd varchar(20) Default 'cairo',

hirdate date Default getdate,

dnum int,);

ADD NEW COLUMN - int;

ALTER TABLE emp ~~add~~ sal int;

Select * From emp;

Delete the column in [Table];

ALTER TABLE emp ~~drop~~ COLUMN sal;

Select * From emp;

Delete the Table. DropTable emp;

DropTable emp;

DML

Insert-Update-Delete.]

Select * From Employee;

Insert into Employee

values(11, 'Billie', 'Eilish', 'F', '2001-12-18',

666, 1);

- Select * From Employee;
- Update Employee ✓
Set DNUM = 666 ✓
Where FIRSTN = 'Madonna'; ✓
- Select * From Employee;
- Insert into Employee
VALUES(12, 'Ava', 'Max', 'F', '1994-2-16', 666, 1);
- Select * From Employee; ✓
• Update Employee ✓
Set FIRSTN = 'Maya', LASTN = 'Diaz' ✓
Where SSN = 11; ✓
- Delete Row
Delete From Employee
Where SSN = 11;
- Select * From Employee;
- Select *
From Employee
Where SSN = 3 and DNUM = 666;
- Select * ✓
From Employee ✓
Where SSN between 1 and 7; ✓
- Select * From Employee; ✓
- Select * From Employee ✓
Where FIRSTN in
('Demons', 'Madonna', 'Ariana', 'Katy'); ✓

Database Schema Design (Relationships and Tables):- Database Diagrams.

1. Relationships (Connect Foreign Key).
2. Tables (with in SQL Server and query code).

The CREATE TABLE statement is used to create a new table in a database.

CREATE TABLE table_name(
 column1 datatype,
 column2 datatype,
 column3 datatype,

)

CREATE TABLE NewTableName
AS
 SELECT column1, column2, ...
 FROM tableName
 WHERE;

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

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

Make sure the order of the values is in the same order as the columns in the table.

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

* return all columns of the queried tables. SELECT *,

BUSINESS INTELLIGENCE (BI) CAREER ROADMAP ITI.

BI PROCESS:-

1. DATA ARCHITECT: Data Warehouse Design.
2. DATA ENGINEER: Data Integration.
3. DATA REPORTING: Data Visualization Developer.
4. DATA ANALYSIS: Data Analyst.

* DATA ANALYST: USED TOOLS.

→ MICROSOFT:-

1. AZURE ANALYSIS SERVICES.
 2. MICROSOFT SQL SERVER.
 3. MICROSOFT EXCEL.
- . SKILLSET :- DAX, MDX.
- . PYTHON JUPYTER NOTEBOOK.

TYPE OF JOINS:-

- * 1. CROSS JOIN.
 - 2. CARTESIAN PRODUCT.
 - * 1. INNER JOIN.
 - 2. EQUIJOIN.
 - * 1. OUTER JOIN.
 - 2. LEFT OUTER JOIN.
 - 3. RIGHT OUTER JOIN.
 - 4. FULL OUTER JOIN.
 - * SELF JOIN.
- (UNARY RELATIONSHIP).
- * UNARY RELATIONSHIP.



(201)

Joins to Merge Tables has same
ID in same Row.
Merge PrimaryKey with
ForeignKey in same Table.

use School;
Insert into Student
values(1, 'Ahmed', 20, 1),
(2, 'Leo', 10, 1),
(3, 'Mars', 30, 1),
(4, 'Khai', 40, 1),
(5, 'Michael', 50, 1),
(6, 'Mike', 60, 1);

SELECT * From Student;

SELECT * From DePT;

// select all columns from table Student.
SELECT Sname, Dname

(From student CROSS JOIN DePT;

↳ ~~using Dept & Student Id is 201.~~

then SELECT columns Sname & Dname.

to merge table has same ID
in same Row.

SELECT student.Did, Sname, student.Did,
Dname

From STUDENT INNER JOIN DePT
ON STUDENT.Did = DePT.Did;

SELECT *
From DePT LEFT JOIN student
ON DePT.Did = STUDENT.Did;

• SELECT *

From Dept RIGHT JOIN Student
ON Dept.Did = Student.Did;

• SELECT *

From Dept FULL OUTER JOIN Student
ON Dept.Did = Student.Did;

• SELECT *

From Dept D1, student S1
where D1.Did = S1.Did
order by D1.Did;

• Select student.sid From student
UNION

Select Dept.Did From Dept;

DB Design

ERD

Normalization

MAPPING

A → B.

SSN → Ename.

Pnum → ProLocat.

A → B, C, D.

PK → Columns.

Sid → Name, Age, address.

H, M → Z. Sid, Cid → Grade.

Fid, Pid → Hours.

Steps in Normalization:-

- * Table with Multivalued Attributes
- Remove multivalued attributes
- * First Normal Form
- Remove Partial Dependencies
- * Second Normal Form
- Remove Transitive Dependencies
- Third Normal Form.

normalization is :-

a technique of organizing the data into multiple related tables, to minimize Data Redundancy. ✓
بيانات مترابطة.

- * Unnecessary data repetition
- increase the size of the Database.
- And Leads to more issues.
- البيانات المترابطة.

1st Normal Form.

Every Table in your Database should at least follow the 1st Normal Form. Always. or stop using Database.

FULL LECTURE. جوال ٢٠
normalization.

Aggregate Function.

[Count-Max-Min-Avg-Sum.] ✓

- Aggregate Function.

(Count, Max, Min, Avg, Sum.)

→ SELECT SUM(Salary)

62500

From employee

→ SELECT MIN(Salary), MAX(Salary)

From Employee

1000 9000 Row

→ SELECT COUNT(eid)

From Employee

15

COUNT(*) → 15

COUNT(eid) → 15

COUNT(ename) → 14

→ SELECT AVG(Salary)

From employee

✓ SELECT MIN(Salary), did

✓ From employee

✓ GROUP BY did;

1000	10
2000	20
1500	30

SELECT COUNT(eid), Address

From employee

GROUP BY ADDRESS;

6 Cairo

5 Alex

4 Mansoura

→ Select COUNT(eid), address

From employee

Where did in (10, 30)

GROUP BY ADDRESS;

3	Cairo
3	Alex
4	Mansoura

PAGE _____
DATE _____

SELECT MIN(Salary), did
FROM Employee
WHERE address like '%_0%'

GROUP BY did;

2000	10
2000	20
1500	30

SELECT COUNT(cid), Address
FROM Employee
GROUP BY Address;

6	Cairo
5	Alex
4	Mansoura

Eid Ename Salary Address Did

1	Ahmed	3000	Cairo	10
2	Ali	5000	Cairo	10
3	Eman	2000	Cairo	10
4	Khalid	1000	Alex	10
5	Yousef	4000	Alex	10
6	Sameh	5000	Alex	10
7	Mohamed	6000	Alex	20
8	Aiaa	7000	Alex	20
9	Ola	4000	Cairo	20
10	Reem	2000	Cairo	20
11	Nada	9000	Cairo	20
12	Sayed	8000	Mansoura	30
13	Reham	1500	Mansoura	30
14	Sally	2000	Mansoura	30
15	Amira	3000	Mansoura	30
		NULL		

```
use SD;
create Table Student
Eid int Primary Key,
Ename varchar(50),
Salary int,
Address varchar(50),
Did int;

insert into Student
values
(1, 'Ahmed', 3000, 'Cairo', 10),
(2, 'Ali', 5000, 'Cairo', 10),
(3, 'Eman', 2000, 'Cairo', 10),
(4, 'Khalid', 1000, 'alex', 10),
(5, 'Yousef', 4000, 'alex', 10),
(6, 'Sameh', 5000, 'alex', 10),
(7, 'Mohamed', 6000, 'alex', 20),
(8, 'Aida', 7000, 'Alex', 20),
(9, 'Olaa', 4000, 'Cairo', 20),
(10, 'Reem', 2000, 'Cairo', 20),
(11, 'Nada', 9000, 'Cairo', 20),
(12, 'Sara', 8000, 'mansoura', 30),
(13, 'Rana', 1500, 'Mansoura', 30),
(14, 'Sally', 2000, 'mansoura', 30),
(15, 'Nur', 3000, 'mansoura', 30);
```

✓ Select sum(salary). did
✓ From employee
✓ Group by Did
✓ Having sum(salary) >= 22000;
→

28000	20
-------	----

✓ Select count(eid), address
✓ From employee
✓ Group By address
✓ Having count(eid) = 5;
→

6	Cairo
5	Alex

SELECT sum(salary). did
From employee
Group By Did
Having count(eid) = 5;

20000	10
-------	----

✓ SELECT sum(salary). did
✓ From employee
✓ Where address like '_a%'
✓ Group By Did
✓ Having sum(salary) > 12000;

15000	20
14500	30

✓ Select max(salary). address
From employee
Where Did in (10, 30)
Group By Address
Having count(eid) > 3;

8000	Mansoura
------	----------

Select $\sum(\text{Salary})$, dePT_id
From INSTRUCTOR
GROUP BY dePT_id;

* Select $\sum(\text{Salary})$, d.dePT_id,
dept_name
From INSTRUCTOR i inner join
Department d
on d.Dept_Id = i.Dept_Id
GROUP BY d.dePT_id, deptname;

→ Subqueries. ✓

✓ Select *

✓ From STUDENT

Subquery

Where st_stage < (Select avg(stage)
From STUDENT);

→ Subquery + DML.

Delete From STUD COURSE
Where ST_ID = 1;

.DELETE From STUD COURSE

.Where ST_ID = (SELECT ST_ID From STUDENT
Where ST_ADDRESS = 'Cairo');

UNION ALL

UNION INTERSECT EXCEPT
Select ST_FNAME
From STUDENT

UNION ALL

Select INS_NAME
From INSTRUCTOR

UNION ALL
UNION
INTERSECT
EXCEPT

→ Print in same column

* Must have datatype.

2024/8/24 02:15

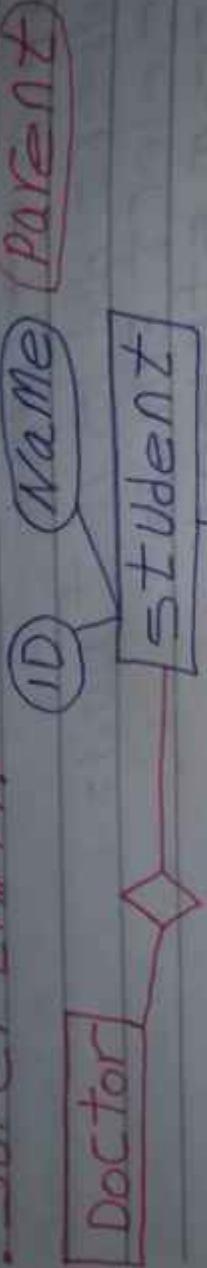
Average of Column Salary

- Count(*) All Columns Number
- Use SD;
- Select COUNT(*), COUNT(Eid),
COUNT(Ename), COUNT(Salary)
From student;
- SELECT avg(Salary)
From student;
- SELECT SUM(Salary) / COUNT(Salary)
From student;
- SELECT SUM(Salary), Did
From student
Where Salary > 1000
GROUP BY Did;
- SELECT SUM(Salary), Did
From student
Where Salary > 1000
GROUP BY Did
having COUNT(Did) < 6;
- Use SD;
- Select *
From student
Where salary < (Select avg(Salary)
From student);
- SELECT *
From student
Where Eid < 7;
- Select *;
Select COUNT(Eid)
From student) AS CountofID
- Select DISTINCT(Salary)
From student
Where Salary IS NOT NULL;

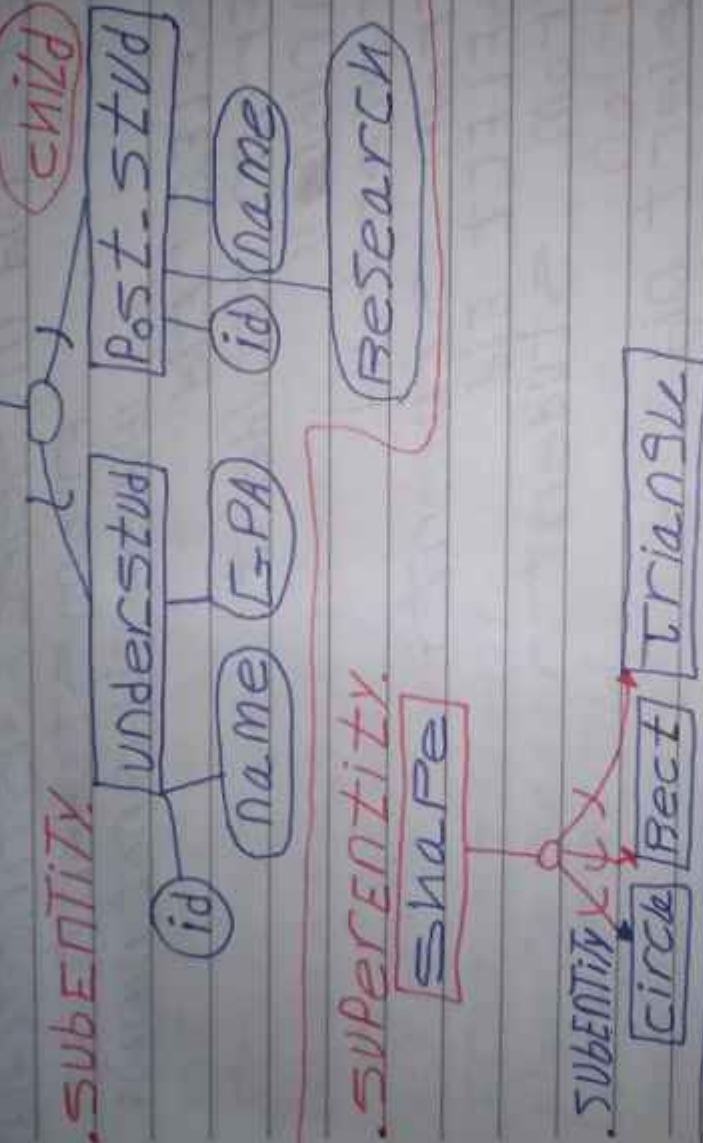
- SELECT distinct(did)
From student
Where did is not null;
- Select Ename
From student
Where did in (SELECT distinct(did)
From student
Where did is not null);
- SELECT Eid
From student
UNION ALL
Select did
From student;
- SELECT Eid
From student
UNION
Select did
From student;
- Select Eid
From student
INTERSECT
Select did
From student;
- Select Eid
From student
EXCEPT
Select did
From student;

EERD (ERD) + Inheritance.

SUPERENTITY.



child



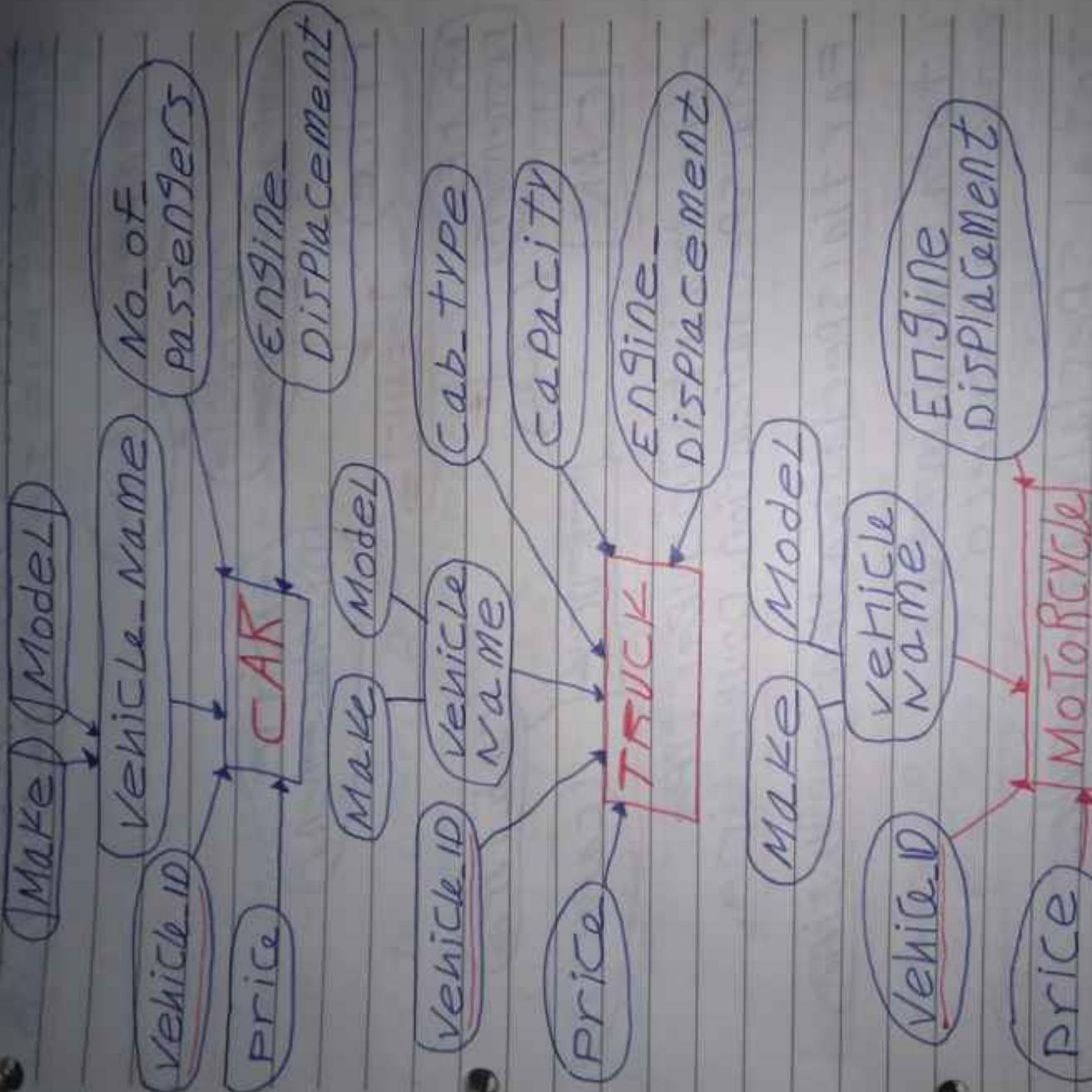
* **SubTYPE** ENTITIES inherit values
of **all** attributes **OF**
the 'superType'!

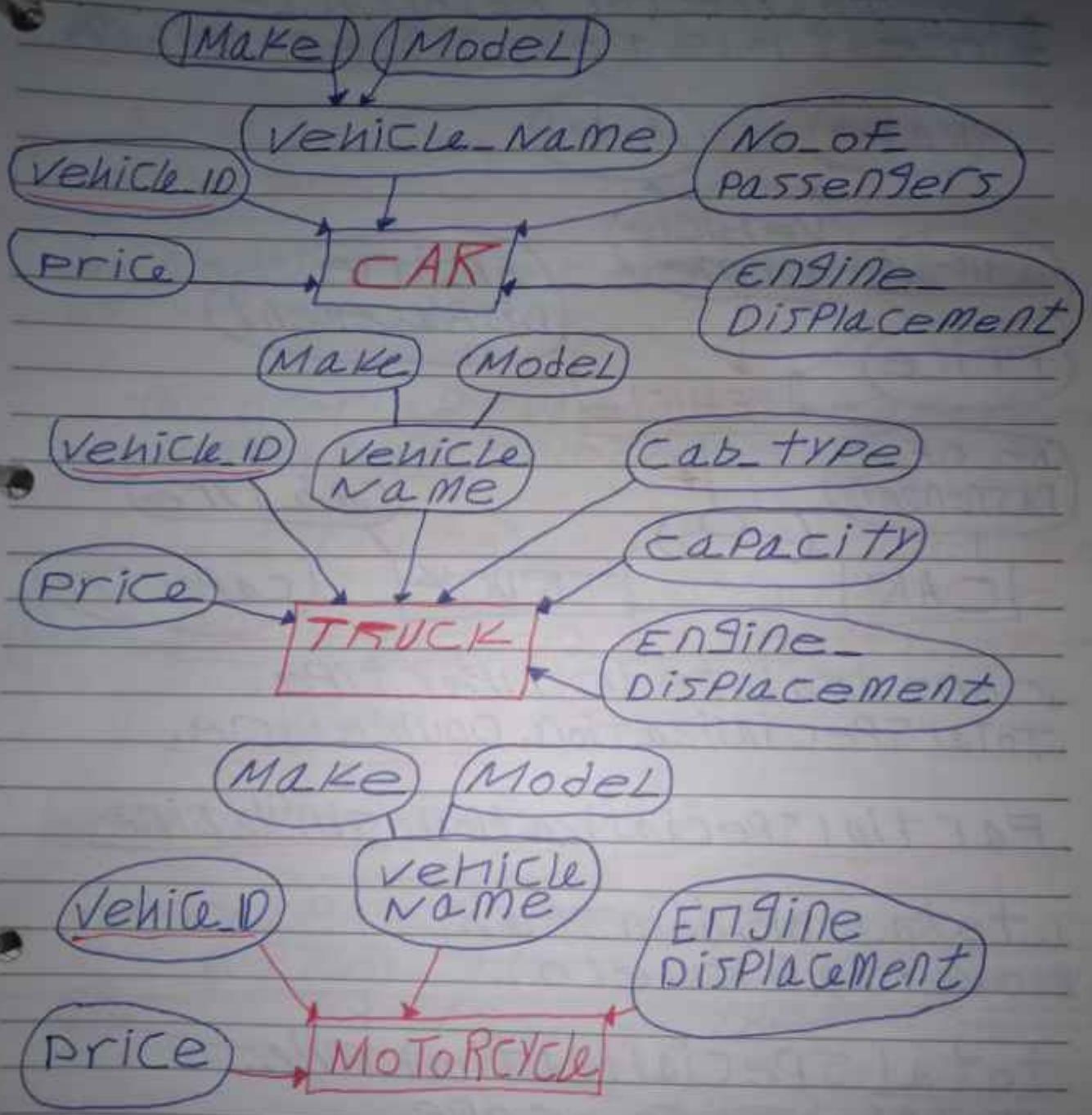
Example of Generalization.

② **Three entity types :-**

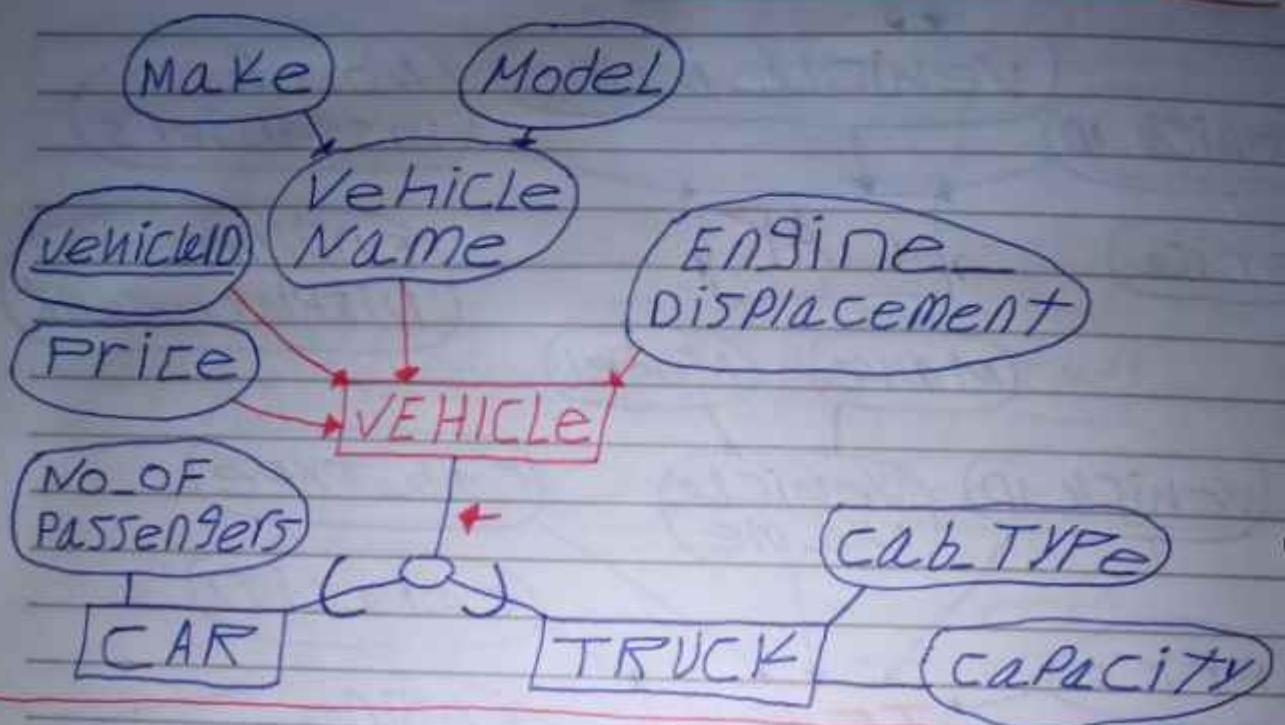
CAR
TRUCK
MOTORCYCLE

* All these types of vehicles
have common attributes.





→ No subtype for Motorcycle,
since it has no unique attributes.



→ constraints in supertype.

• TOTAL Specialization. Double Line.

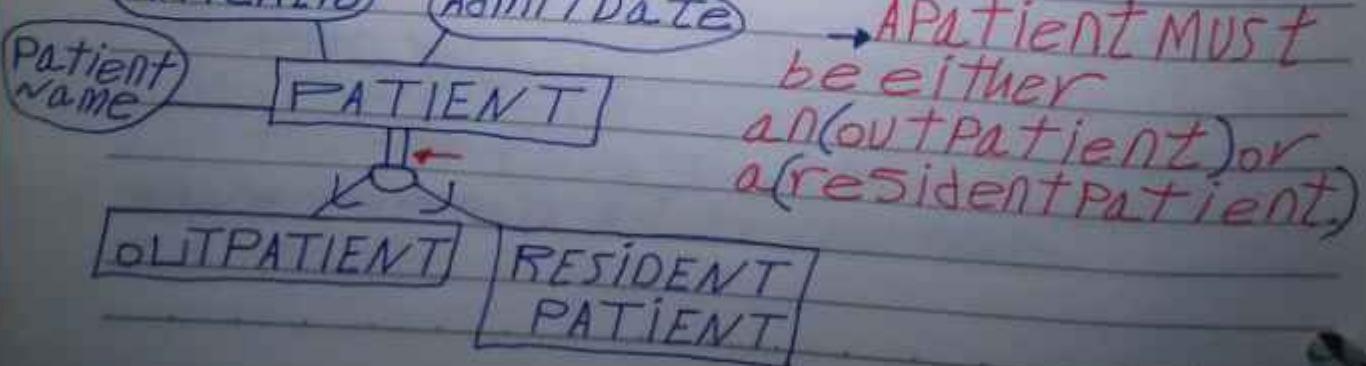
• Partial Specialization. Single Line.

1. TOTAL Disjoint(d).

2. OVERLAP RULE(o).

• TOTAL Specialization rule.

• TOTAL → zero or one.

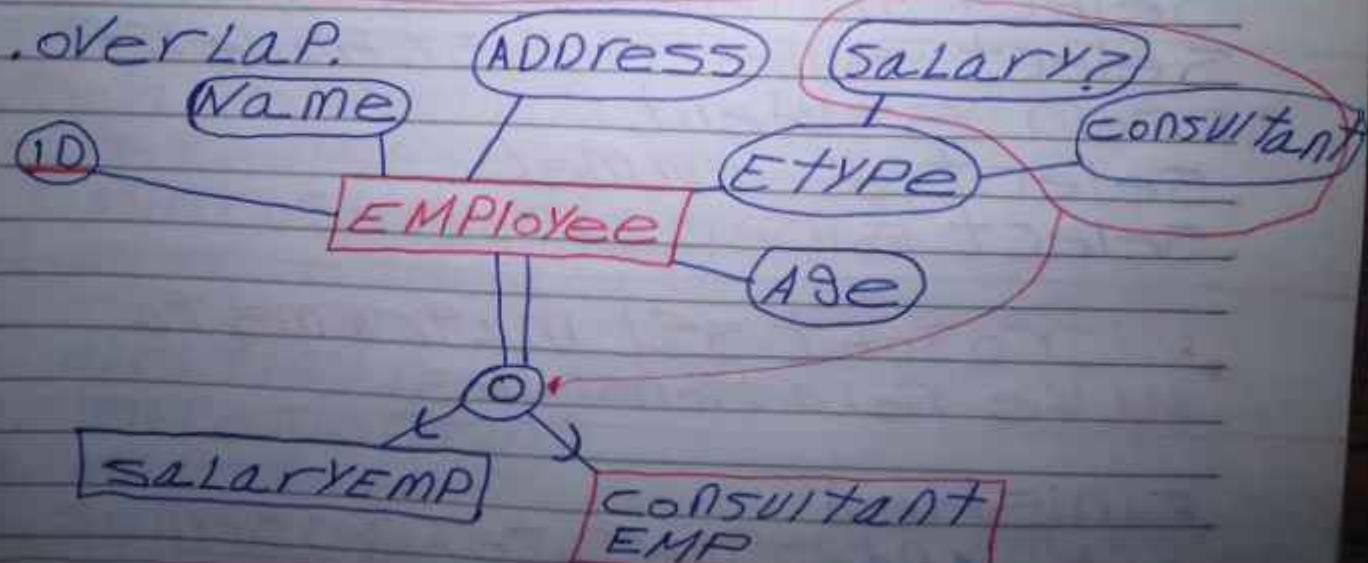


Partialspecialization rule.

→ A vehicle could be a car, a truck, or neither.

the disjoint rule can only be a member of one subtype. (d).

the overlap rule can be a member of multiple subtypes. (o).



→ `SELECT st_Fname, st_Age, dept_id
From student
order by dept_id asc, st_Age desc`

→ `delete from Department where
dept_id = 20;`
→ `update Department set
dept_id = 4000 where dept_id = 20;`

BUILTIN FUNCTIONS

.AGG FUNCTIONS:-

`getdate()` `isnull` `coalesce`
`concat` `convert`

`SELECT Year(getdate());`

`SELECT Month(getdate());`

`SELECT Substring(st_Fname);`

`Select month(getdate());`

`Select Substring(st_Fname, 1, 3)`
From student

`Select db_name()`

`Select SUSERNAME()`

→ Microsoft SQL Diagrams to
MAKE RELATIONSHIPS.

.Finished

EERD and Aggregate Function.

'Microsoft PowerBI Data Analyst'
Professional certificate.

→ Power BI ANALYST. COURSERA.

→ SKILLS You'll gain:-

1. DATA ANALYSIS.
2. MICROSOFT EXCEL.
3. SQL.
4. PowerBI.
5. PowerQuery.

* Prepare for a career in

→ BUSINESS intelligence.

→ EXAM PL-300: Microsoft

Power BI Data Analyst.

→ PROFESSIONAL level Training From Microsoft.

* QUALIFY for in demand job titles:-

1. BUSINESS intelligence Analyst.
2. BUSINESS Data Analyst.
3. Power BI Analyst.

. Relational Database Management Systems:-

1. MICROSOFT SQL Server.
2. oracle database.

1. Writing queries.

2. Implementing.

3. Maintaining.

4. SQL Server Business Intelligence.

→ SQL Server.

SQL Server FULLY RDBMS.

• tables one to many relationships.

Data quality service & master data service.

• Services (Local).

→ DB Engine SQL Server (MSSQL Server).

Application:-

-- SQL Server Management Studio.

. Setup (instance):-

1. Services:-

1. DB Engine

SQL Server (MSSQL Server).

2. SSIS

SSRS BI

SSAS

3. Data Quality Service.

Egypte.

Egypt.

Egypt.

APPLICATION:-

1. SQL Server Management Studio.

2. SQL Server Data Tools.

3. Data Quality Client.

4. SQL Server Profiler.

5. SQL Server Tuning Advisor.

SQL Server

Server type:-

1. Database Engine.
2. Analysis Services.
3. Reporting Services.
4. Integration Services.

→ Database Engine configuration.
→ Authentication Mode.

→ SQL Server authentication and
Windows authentication.

→ Server Properties.

→ SQL Server and Windows Authentication
Model.

→ Authorization (Permissions).

→ Authentication (username & Password).

→ Windows authentication
Win Admin → SQL Admin.

→ SQL Server authentication.

Dev Ahmed 123.

Dev Ali 444.

→ Server Properties :- SQL Server.
Security.

→ SQL Server and Windows Authentication
mode.

→ Login auditing.

• Failed Logins ONLY.

SECURITY

. Logins.

1. → Login - New.

2. → LoginName: Dev ALI

• SQL Server Authentication.

Password: ...

CONFIRMPASSWORD: ...

OK

→ SQL Server.

→ Databases.

ITI.

+ SECURITY

→ USER DatabaseUser.

1. User Name:-

Dev ALI

2. Login_name

Dev ALI

OK

Logins.

>Login Properties.

Permission to connect to
Database Engine :-

• GRANT

• DENY

Login:

• ENABLED

• DISABLED

→ OK.

- EXCEL → from other sources →
From SQL Server.
- Create a connection to SQL Server table.
- Connect to Database server.
 1. Servername
 2. Logon
- Use following User Name and Password.
Username: SA
Password: ...
- Select database and Table.
- Next.
- Import Data.
- Table.
- OK

SQLCMD.

```
C:\Users\RAMI>sqlcmd -S "..."  
use ITI  
go  
changed database context to 'ITI'.  
creating table mytesting(id int, name varchar(10))  
go
```

```
C:\Users\RAMI>sqlcmd -S "..."  
use ITI  
go  
changed database context to 'ITI'
```

>Create table mytesting(Id int,
name varchar(20))

:go

:select * from course

:go

:exit

C:\Users\RAMI\59\cmd-5

"." -d "ITI" -q "Select * from topic"

→ Top ID . Top Name

1. Programming

2. DB

3. Web

4. operating system

5. Design

→ exit

C:\Users\RAMI\59\cmd-5". "-d

"ITI" -q "Select * from topic"

-o "d:\myITIResult.txt"

Save it to FILE text.

→ Select top(3) *
From student;

TOP(3)

→ SELECT top(3) st_fName
From student;

→ SELECT top(3) *
From student;

- `SELECT TOP(3)*
From STUDENT
Where ST_ADDRESS = 'alex';`
- `SELECT TOP(3) SALARY
From INSTRUCTOR;`
- `SELECT MAX(SALARY)
From INSTRUCTOR;`
- `SELECT TOP(3) SALARY
From INSTRUCTOR
order by SALARY DESC;
100000
20000
15000`
- `SELECT TOP(2) DISTINCT SALARY
From INSTRUCTOR
order by SALARY DESC;`
- `SELECT TOP(4) WITH TIES*
From STUDENT
order by ST_AGE;`
- `ST_AGE`
 - `NULI`
 - `20`
 - `21`
 - `21``SELECT TOP(7) WITH TIES*
From STUDENT
order by ST_AGE;`
- `SELECT NEWID() -- GUID`
- `Select *, NEWID()
From STUDENT;`

Random Number Address Memory.

• Select * , newid()
From STUDENT;
• SELECT TOP(3)*
From STUDENT RNG
order by Newid()
• SELECT FNAME + '' + LNAME AS FULLNAME
From STUDENT
order by FULLNAME;
• SELECT FNAME + '' + LNAME AS FULLNAME
From STUDENT
Where FNAME + '' + LNAME = 'ahmed ali';
• Select *
From ((SELECT FNAME + '' + LNAME
AS FULLNAME) From STUDENT
AS NewTable)
Where FULLNAME = 'ahmed ali';

Execution order :-

1. From ✓
2. Join ✓
3. On ✓
4. Where ✓
5. Group ✓
6. Having [agg]. ✓
7. Select [distinct + agg]. ✓
8. Order By ✓
9. Top ✓

• DBObjects [table view function
SP RULE].

• [ServerName].[DBName].
[SchemaName].[ObjectName]

• SELECT *
From company_SD.dbo.Project
• Select dname
From companySD.dbo.Departments
UNION ALL
Select dept_name
From Department

DDL

Select * into table2
From student; COPY PAST

SELECT * into company_Sd.dbo.student
From student;

Select ID, Fname into tab3
From student
Where Address = 'ALEX';

Select * into tab4
From student
Where stage < 1000;

Insert into tab3
VALUES(66, 'ALI');

→ Insert based in select.
→ Insert into tab3
Select id, Fname
From student;

PAGE _____
DATE _____

```
SELECT SUM(Salary)
FROM Instructor
HAVING COUNT(id) > 100;
```

Ranking Functions.

- Row_Number()
- Dense_Rank()
- NTiles(Group)
- Rank()

	eid	ename	esal	did	RN	DR	NTILE
	15	Ahmed	10000	10	1	1	1
	14	Ali	10000	10	2	1	1
	12	Eman	9000	10	3	2	1
	1	Nada	9000	10	4	2	1
	2	Reem	9000	10	5	2	1
	3	Khalid	8000	10	6	3	2
	7	Mohamed	7000	20	7	4	2
	8	Sayed	7000	20	8	4	2
	6	Hassan	6000	20	9	5	2
	5	Omar	6000	20	10	5	2
	9	Sally	5000	30	11	6	3
	10	Shimaa	4000	30	12	7	3
	11	Nana	4000	30	13	7	3
	12	Lama	3000	30	14	8	3

```
SELECT *, Row_Number()
OVER(ORDER BY esal DESC) AS RN,
Dense_Rank()
OVER(ORDER BY esal DESC) AS DR
FROM employee;
```

```
SELECT *
FROM (
    SELECT *, RowNumber()
    OVER(ORDER BY eSAL DESC) AS RN,
    Dense_Rank()
    OVER(ORDER BY eSAL DESC) AS DR,
    NTILE(3) OVER(ORDER BY eSAL DESC) AS G
    FROM Employee) AS NewTable
WHERE RN = 1          DR = 1
      RN = 3          DR <= 2
      RN >= 2          G = 1
```

→ INSERT INTO

```
SELECT *
FROM (
    SELECT *, RowNumber()
    OVER(PARTITION BY did ORDER BY eSAL DESC) AS RN,
    Dense_Rank() OVER(PARTITION BY
        did ORDER BY eSAL DESC) AS DR
    FROM Employee) AS NewTable
WHERE RN = 1          DR = 1
      RN = 3          DR <= 2
```

```
. SELECT *, RowNumber()
    OVER(ORDER BY ST_Age DESC) AS RN
    FROM Student;
. SELECT *, Dense_Rank()
    OVER(ORDER BY ST_Age DESC) AS DR
    FROM Student;
```

```
Select *  
From (  
    Select *, Row_number()  
    over (order by st_Age desc)  
    AS RN  
    From student) AS NewTable  
Where RN = 1;
```

DATA TYPES:-

1. NUMERIC DT.
2. DECIMAL DT.
3. CHAR DT.
4. DATETIME.
5. BINARY DT.

Data Types

- NUMERIC DT
- bit bool 0:1 true:False
- tinyint 1Byte -128:+127 Unsigned 0:255
- smallint 2B -32768:+32767 Unsigned 0:65535
- int 4B
- bigint 8B

→ DECIMAL DT

Char DT

DATETIME

BINARY DT

DECIMAL DT.

SMALLMONEY 4B .0000

MONEY 8B .0000

REAL .000000

Float .000000000000000000
dec decimal dec(5,2)

→ CharDT.

Char(10)

Varchar(10)

□ Char(10)

□ Varchar(10)

□ Varchar(MAX)

→ Datetime

Date MM/DD/YYYY

Time HH:MM:12.765

time(7) HH:MM:12.7659876

Smalldatetime MM/DD/YYYY HH:MM:00

datetime MM/DD/YYYY HH:MM:SS.987

datetime2(7) MM/DD/YYYY

datetimeoffset 11/24/2020 10:30

+2:00 Timezone

→ binaryDT

binary 0111100 1111100

Datetime MM/DD/YYYY HH:MM:SS.987

Datetime2(7) MM/DD/YYYY

Datetimeoffset 11/24/2020 10:30

+2:00 Timezone

binaryDT

binary 0111100 1111100

image

others

.XML

.unique_identifier

.SQL variant

DB Engine

- Types of instances
- Types of Authentication
- ToP
- SELECT into
- Ranking
- Date type

SELECT ins_name, salary
FROM Instructor

SELECT ins_name, salary,

* CASE

When salary >= 3000

then 'HighSal'

When salary < 3000

then 'Low'

ELSE 'No Value'

* END AS Newsal

FROM Instructor;

Update Instructor

set salary =

CASE

When salary >= 3000

then salary * 1.10

ELSE salary * 1.20

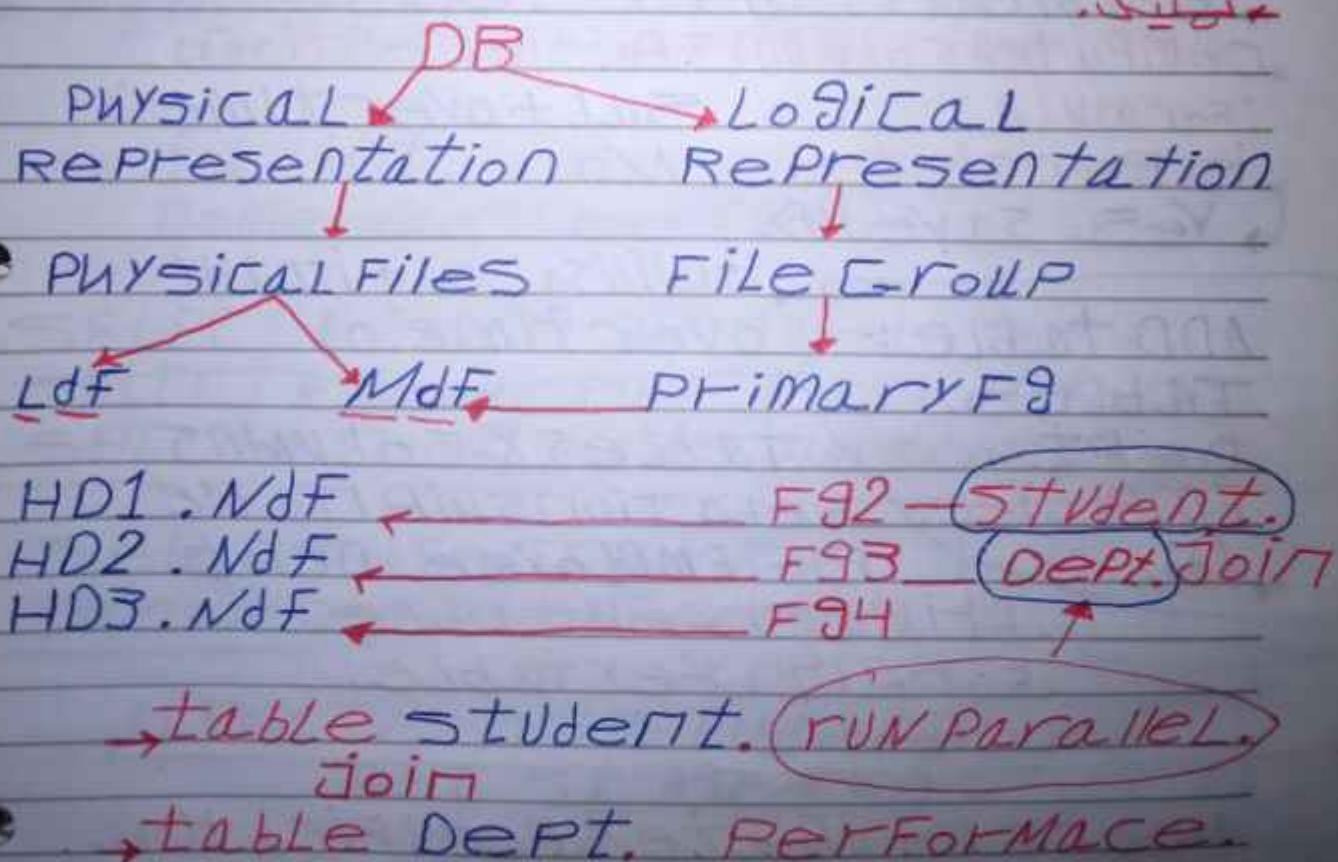
END

→ SELECT name, iif(salary >= 3000,
'High', 'Low')

From Instructor;

• SELECT cast(getdate() as varchar(20))
 • SELECT convert(varchar(20),
 getdate(), 102)
 • Select convert(varchar(20),
 getdate(), 105)
 • Select Format(getdate())
 • Select Format(getdate(), 'dd/MMMM
 yyyy')
 • Select Format(getdate(),
 'ddd MMM yy')
 • Select Format(getdate(), 'dd'))

SELECT eomonth(getdate())
 SELECT Format(eomonth(getdate()), 'dd')
 SELECT eomonth('1/1/2000');



Create - Creation Database.

PARALLEL PROCESSING.

→ HardDisk1 Parallel HardDisk2.

→ DIFFERENT PHYSICAL FILES.

HD3 → 32 Kb. Data Pages.

→ NEW DATABASE.

→ LOGICAL NAME :-

DB1.MDF

DB1_Log.LDF

FILE2.NDF

FILE3.NDF

FILE4.NDF

→ CREATE TABLE THEN.

SET PRIMARY KEY.

→ COLUMN PROPERTIES.

COMPUTED COLUMN SPECIFICATION.

(FORMULA) SAL + overtime.

IS PERSISTED NO.

→ YES. SAVE HD.

ADD TABLE :-

TABLES.

DEPT.

EMPLOYEE.

Add.

RELATIONSHIP NAME

FK_EMPLOYEE_DEPT

PRIMARY KEY TABLE.

FOREIGN KEY TABLE.

OK

ISNULL(SAL,0) + ISNULL(OVERTIME,0)

INSERT AND UPDATE SPECIFICATION.

DELETE RULE: NO ACTION - CASCADE

SETNULL - SETDELETED

CASCADE: Delete or update an entry from both the child and parent table.

- SchemaName.ObjectName.
- SELECT *
FROM dbo.Student;
- CREATE SCHEMA HR;
- CREATE SCHEMA SALES;
→ Table to Schema.
- ALTER SCHEMA HR TRANSFER STUDENT;
- ALTER SCHEMA HR TRANSFER INSTRUCTOR
- ALTER SCHEMA SALES TRANSFER
DEPARTMENT
- Schema → External Tables.
CREATE TABLE STUDENT
(
id int,
name varchar(20));
- SELECT * FROM INSTRUCTOR;
- SELECT * FROM HR.INSTRUCTOR;
- SELECT * FROM HR.STUDENT;
- Select users or roles.
→ Browse for objects.
- UPDATE HR.STUDENT
SET ST_Age += 1;

~~DELETE FROM hr.student
Where st_id = 666;~~
~~Short cut.~~

Create synonym HE
For
HumanResources.
EmployeeDepartmentHistory;
→ SYNONYMS.
Create synonym schema.

Alter synonym HE
For HumanResources.
EmployeeDepartmentHistory;
· SELECT * FROM HE;

Drop table course; Data & Metadata.
Delete from course; Data.
truncate table course; Data.

Create table Deptl
Dept_ID int primary key,
DName varchar(20),);

Create table hr.emp[
eid int identity(1,1),
ename varchar(20),
eadd varchar(20) DEFAULT 'alex',
hiredate date DEFAULT getdate(),
sal int,
overtime int,
netsal as (ISNULL(sal, 0) +
ISNULL(overtime, 0)) Persisted

• BD date.

age as Year(GetDate()) - Year(BD),

Gender varchar(1),

HourRate int not null,

Did int,

Constraint C1 Primary Key(Did, EnName),

Constraint C2 Unique(Sal),

Constraint C3 Unique(Overtime),

Constraint C4 Check(Sal > 1000),

Constraint C5 Check(LeadIn
(Cairo, Mansoura, Alex)),

Constraint C6 Check(Gender = 'F' or
Gender = 'M'),

Constraint C7 Check(Overtime
Between 100 and 500),

Constraint C8 Foreign Key(Did)

References Dept(Dept_id)

on Delete Set Null on Update Cascade);

• ALTER TABLE EMP add constraint

C100 CHECK(HourRate > 100);

• ALTER TABLE EMP drop constraint C3;

Constraint → New Data. (جديد)

Constraint → Shared.

Data Type.

Rule.

PK

CHECK

NULL, NOT NULL

→ RULE on COLUMN

ALTER TABLE INSTRUCTOR ADD
CONSTRAINT C200
CHECK(SALARY > 1000);

RULE

→ CREATE RULE r1 AS @X > 1000
SP BIND RULE r1, 'INSTRUCTOR.SALARY'
SP BIND RULE r1, 'EMP.OVERTIME'

- BIND RULE برمجة الـ Rule.
- UNBIND RULE إلغاء برمجة الـ Rule.

• SP UNBIND RULE 'INSTRUCTOR.SALARY'.
• SP UNBIND RULE 'EMP.OVERTIME'.
• DROP RULE r1
→ DEFAULT VALUE.
• CREATE DEFAULT DEF1 AS 5000
• SP BIND DEFAULT DEF1,
'INSTRUCTOR.SALARY';
→ DROP DEFAULT DEF1;

→ CREATE DATATYPE COMPLEXDT
(INT > 1000 DEFAULT 5000).

→ CREATE RULE r1 AS @X > 1000;
→ CREATE DEFAULT DEF1 AS 5000;
→ SP ADD TYPE COMPLEXDT, 'INT';
→ SP BIND RULE r1, COMPLEXDT;
→ SP BIND DEFAULT DEF1, COMPLEXDT;
→ CREATE TABLE test3C
ID INT,
NAME VARCHAR(20),
SALARY COMPLEXDT);

→ SQL Constraint.

- Constraint C1 Primary Key (eid, ename),
- Constraint C2 Unique (sal),
- Constraint C4 Check (sal > 1000),
- Constraint C5 Check lead in ('cairo', 'mansoura', 'alex')),
- Constraint C7 Check (overtime between 100 and 500),
- Constraint C8 Foreign Key (did) References dePT(dePT_id) on Delete set NULL on Update cascade)

→ Union - union all - intersect EXCEPT.

- Himalaya college of Engineering.
- SQL TUTORIAL.
- TUTORIALSPoint.com.
- Python SQL tutorial.
- technical Documentation.
- Microsoft SQL Documentation.

• Metadata : is Descriptive of Data.
البيانات الوصفية هي بيانات وصفية.

• Metadata (or MetaInformation) About other data.

• Data that Provides information About other data.

البيانات التي توفر معلومات حول البيانات الأخرى.

→ SQL constraint.

- Constraint C1 Primary Key (eid, ename)
- Constraint C2 Unique (sal),
- Constraint C4 check (sal > 1000),
- Constraint C5 check (dept_name in ('Cairo', 'mansoura', 'alex')),
- Constraint C7 check (overtime between 100 and 500),
- Constraint C8 Foreign key (did) references dept(dept_id) on delete set null on update cascade)

→ Union - union all - intersect EXCEPT.

- Himalaya college of Engineering.
- SQL TUTORIAL.
- TUTORIALSPoint.com.
- PythonSQL tutorial.
- technical Documentation.
- MICROSOFT SQL Documentation.

• MetaData : is Descriptive of Data.
البيانات الوصفية هي بيانات وصفية.

- Metadata (or Metainformation) About other data.
- Data that provides information About other data.

البيانات التي توفر معلومات حول البيانات الأخرى.

✓ SQL Day 7. ADAMS AL DAHABY.

SQL Day 8. ادامرزال دهابي.

SQL Day 9.

SQL Day 10.

→ ITI BUSINESS INTELLIGENCE TRACK.

• MICROSOFT SQL Server.

• MICROSOFT Power BI.

• ADAMS AL DAHABY. WORLD IS YOURS.

→ USING WINDOW CLOCK.

→ Get Start And Basic.

→ Search by Yourself in Program
AS SQL Server

→ Explore Program By Yourself.

• SQL Days 1h → 1h 6 min. important.

→ SQL Procedures.
Triggers.
XML.

→ SQL Procedures Triggers XML.
→ Definition.

• ASCII TABLE. جدولات حروف.

• virtual Table.

• View is virtual table not
a table.

• Target and Source

• Merge into SQL

• Procedures similar Function

• MAKE FUNCTION and View to
use in Procedure.

2024/8/25 14:06

عالم زكاء أصطناعي

المهندس أداة التهبي.

- watch video. then study it.
- write notes. solving problems.

عالم زكاء أصطناعي. علم يحاكي الإنسان البشري.
الذكاء الاصطناعي آخر صنف الإنسانية **MICROSOFT**.

والمعلم.

• Ariana Grande songs LYRICS.

• Genius lyrics, songs, albums.

• SQL Server For interviews

• TOP SQL Server interview questions and answers.

Variables:-

1. LOCAL variables

2. GLOBAL variables

→ LOCAL var:- LOCAL variables.

✓ Declare @X int.

✓ set @X = 10

✓ select @X = 100

✓ select @X = age from student
where ID = 1;

✓ UPDATE Student

set fname = 'omar', @X = age

where ID = 9;

✓ SELECT @X

Variables

• Global Variables

- can't declare Global var
- can't assign Global var

@@

call Global Variable.

@@

SELECT @@servername

SELECT @@rowcount

SELECT @@version

SELECT @@error

SELECT @@identity

@@

servername.

rowcount.

version.

error.

identity.

batch - Function.

Stored Procedure.

use it!

declare @X int ✓

Set @X = 10 ✓

SELECT @X

Result

10

declare @X int = 1000

. Declare @X int = (SELECT avg(st_age)
From student)

. SELECT @X;

→ Assign value to variable X.

Declare @y int

SELECT @y = st_age FROM student
where st_id = 6;

SELECT @y; → 25

SELECT == Print.

Declare @Y int = 100

Select @y = st_age FROM student
where st_id = 9;

SELECT @y;

Declare @y int = 100

SELECT @y = st_age FROM student
where st_address = 'alex';

SELECT @y;

Declare @y int, @name varchar(20)

Select @y = st_age, @name = st_fname
From student where st_id = 9;

SELECT @y, @name;

Declare @Z int

UPDATE STUDENT

Set st_fname = 'ali', @Z = dept_id

where st_id = 7;

SELECT @Z;

.Declare @X int = (SELECT avg(st_age)
From student)

.SELECT @X;

→ Assign value to variable X.

Declare @Y int

SELECT @Y = st_age FROM student
where st_id = 6;

.SELECT @Y; → 25

.SELECT == Print.

.Declare @Y int = 100

SELECT @Y = st_age FROM student
where st_id = 9;

.SELECT @Y;

Declare @Y int = 100

SELECT @Y = st_address FROM student
where st_address = 'alex';

.SELECT @Y;

Declare @Y int, @name varchar(20)

SELECT @Y = st_age, @name = st_fname
FROM student WHERE st_id = 9;

.SELECT @Y, @name;

Declare @Z int

UPDATE STUDENT

SET st_fname = 'ali', @Z = dept_id
WHERE st_id = 7;

.SELECT @Z;

DECLARE @t TABLE(X INT)
INSERT INTO @t
SELECT ST_ID FROM STUDENT
WHERE ST_ADDRESS = 'ALEX';
SELECT COUNT(*) FROM @t;

• Declare @X int = 5
SELECT TOP(@X)*
FROM STUDENT;
→ TOP 5 Rows.

• Declare @Col VARCHAR(20) = '*';
@Tab VARCHAR(20) =
'STUDENT'
SELECT @Col FROM @Tab;
• SELECT * FROM STUDENT;
• SELECT * FROM STUDENT;
• Select @Col FROM @Tab;
• EXECUTE ('SELECT * FROM STUDENT')
• Global variable :-
SELECT @@SERVNAME;
SELECT @@VERSION;
• UPDATE STUDENT
SET ST_ID += 1;
• SELECT @@ROWCOUNT;
• SELECT @@ROWCOUNT;
• SELECT * FROM STUDENT
GO
• SELECT @@ERROR;
• SELECT @@IDENTITY;
→ Number IDE insert.

Global Variables:-

- @@SERVNAME
- @@CONNECTIONS
- @@MAX_CONNECTIONS
- @@CPU_BUSY
- @@ERROR
- @@IDLE
- @@LANGUAGE
- @@TRANCOUNT
- @@VERSION

→ Control of Flow Statement.

if - begin - end - if exists if not exists -
while - constinue - break . case - if
waitfor - choose

→ Control of Flow Statement.

IF

Declare @X int

UPDATE STUDENT

Set st_age += 1;

Select @X = @@ROWCOUNT

IF @X > 0

SELECT 'Multi Rows Affected'

else

SELECT 'No Rows Affected';

create table student(

id int,

name varchar(20));

```
Declare @X int  
UPDATE STUDENT  
    set st age+=1  
SELECT @X = @@ROWCOUNT  
IF @X > 0  
begin  
    Select 'MULTI ROWS AFFECTED'  
end  
else  
begin  
    Select 'NO ROWS AFFECTED'  
end
```

→ Create table student
(id int,
name varchar(20));

→ SELECT * FROM sys.all_columns;

→ built in schema:-

```
SELECT * FROM sys.all_columns;  
SELECT * FROM sys.tables;  
SELECT * FROM sys.all_views;  
SELECT name FROM sys.tables;
```

→ talk to Meta Data.

Select name

From sys.tables

Where name = 'student';

→ students

٢٠٢٤/٨/٢٥ | ji

```
IF EXISTS( SELECT name  
          FROM sys.tables  
          WHERE name = 'Students')  
    SELECT 'Tables is existed'  
ELSE  
    CREATE TABLE Students  
(  
    id int,  
    name varchar(20));  
→ Table is existed.  
Delete From Department  
Where DEPT_id = 20; ✓  
  
IF NOT EXISTS( SELECT dePT_id  
              FROM Student  
              WHERE dePT_id = 20)  
AND NOT EXISTS( SELECT dePT_id  
                  FROM Instructor  
                  WHERE Dept_Id = 20)  
Delete From Department  
Where dePT_id = 20;  
  
.begin try  
    Delete From Department  
    Where dePT_id = 20  
.end try  
.begin catch  
    Select 'error'  
.end catch
```

```
.Begin try  
    Delete From Department  
    Where Dept_id = 20  
End try  
.Begin catch  
    SELECT 'error'  
    Select ERROR_LINE(),  
        ERROR_NUMBER(),  
        ERROR_MESSAGE()  
End catch
```

```
.Begin try  
End try  
.Begin catch  
End catch
```

```
.Begin try ✓  
End try ✓  
.Begin catch ✓  
End catch ✓
```

```
.Begin try End try ✓  
.Begin catch End catch ✓
```

→ While.

```
Declare @X int = 10  
While @X <= 20  
    begin  
        set @X += 1  
        If @X = 14  
            continue  
        If @X = 16  
            break  
        Select @X
```

end

11 12 13 15

العنوان Adam Golden

Begin try
Delete from Department
where Dept_id = 20

End try

Begin catch

SELECT 'error'

Select ERRORLINE(),
ERROR NUMBER(),
ERROR MESSAGE()

End catch

Begin try

End try

Begin catch

End catch

Begin try ✓

End try ✓

Begin catch ✓

End catch ✓

Begin try End try ✓

Begin catch End catch ✓

While.

→ Declare @X int = 10

While @X <= 20

.begin

set @X += 1

IF @X = 14

continue

IF @X = 16

break

select @X

.end

11 12 13 15

```
· Declare @X int = 10
  While @X <= 20
    Begin
      Set @X += 1
      If @X = 14
        Continue
      If @X = 16
        Break
      Select @X
    End
```

11 12 13 14 15

- continue
- break
- case
- iif
- waitfor
- choose

batch-transaction-script.

- insert
- create table
- update
- go
- delete
- drop table

→ create table

go

drop table

go

SP_bindrule

create rule

go

SP_bindrule

Create table

Go

Drop table

Go

Create rule

Go

SP bindrule

.ITI database

.TASKS

.Generate scripts

.Generate Publish scripts

.NEXT

.Set scripting options.

.NEXT

.Save or Publish scripts.

CANCEL NEXT.

→ Save or Publish scripts.

FINISH

transaction [set of queries]

→ single unit of work.

Create table Parent

Pid int Primary Key;

};

Create TABLE Parent

PID int Primary Key;

};

Create table Child

CID int Foreign Key

References Parent(PID);

```
.create table Parent(  
    PID int primary key)  
.create table Child(  
    Cid int foreign key references  
    Parent(Pid))  
.insert into Parent values(1);  
.insert into Parent values(2);  
.insert into Parent values(3);  
.insert into Parent values(4);  
.SELECT * FROM Child;  
.TRUNCATE table Child;  
.TRUNCATE table Child;
```

query SSMS CMD LDF MDF
insert → execute(F5) → begin transaction → OK
commit insert

Delete → Execute begin transaction
Delete → error.
Rollback

logical.

IF Delete:

Go 'OK'

else:

Go Rollback.

begin transaction

insert

Update

Delete

commit-rollback

begin transaction

insert

4Pdate

Delete

~~commit~~ rollback.

begin transaction

insert → ok

update → OK

Restart server.....

Automatic Rollback

begin transaction

insert

Update

delete

commit rollback

Begin transaction

```
insert into child values(1);
```

insert into child values(2);

insert into child values(3);

rollback

الأخوة العبرانيين $\beta\text{-YMg}$.

• Begin transaction what do.
→ Begin transaction
→ Rollback
• Can be rolled back to return the Data to this known state of consistency
مكتنراً (رجاع) إلى حالة الارتباط

• begin transaction
• insert into child values(1);
• insert into child values(5);
• insert into child values(3);
• commit

• Select * from child;
• truncate table child;

• Begin try ✓
• Begin transaction ✓
 insert into child values(1);
 insert into child values(5);
 insert into child values(3);
• Commit ✓

• End try
• begin catch
 rollback ✓
• end catch

→ Begin TRY END TRY.
→ begin catch END catch.
→ begin transaction. Commit if
 rollback

رجاع

FUNCTIONS:-

- 1. BUILT IN FUNCTIONS.
- 2. USER DEFINED FUNCTIONS.

BUILT IN FUNCTIONS:-

- NULL ISNULL COALESCE NULLIF.
- System db_name() user_name().
- convert convert cast format.
- string substring UPPER LOWER len.
- Date getdate() Year Month Day.
- Aggregate COUNT MAX MIN AVG SUM.
- MATH (Power Log Sin Cos Tan).
- ranking FUNCTIONS: Row Number
- Rank Dense_rank Ntile.

LOGICAL FUNCTION iif, choose.

windowing Lead Lag

First_value, Last_value.

String Substring UPPER LOWER len.

- Date getdate() Year Month Day.
- Avg Count MAX MIN AVG SUM.
- MATH Power Log Sin Cos Tan.
- ranking Row Number Rank
- Dense_rank Ntile

Ranking 123.

Ranking 11 22 33.

Grouping 55 4.

FUNCTIONS:-

1. built-in functions.

2. scalar function.

TYPE:-

- NULL system convert string Date

Avg Math ranking ~~Logical Function~~ Windowing

- ISNULL(), COALESCE(), NULLIF(), DB_NAME()
- SUSER_NAME() convert() CAST()
- FORMAT() SUBSTRING() UPPER() LOWER()
- LEN() GETDATE() YEAR() MONTH() DAY()
- COUNT() MAX() MIN() AVG() SUM()
- POWER() LOG() SIN() COS() TAN()
- ROW_NUMBER() RANK() DENSE_RANK()
- Ntile(), IIF() CHOOSE(), LEAD(), LAG()
- FIRST_VALUE() LAST_VALUE()

User Defined Functions:-

Scalar Function

→ Return one value

• inline table function

ReturnTable ---> body ---> select

---> view.

• multi statement table valued

function:-

ReturnTable ---> body ---> select

---> if, declare, while.

→ insert based on select.

• Insert Based

• ON SELECT

FUNCTIONS:-

• ~~SELECT GETDATE()~~
• ~~SELECT ISNULL(ST_FNAME, '')~~
• ~~FROM STUDENT;~~
• ~~SELECT UPPER(ST_FNAME),~~
• ~~LOWER(ST_LNAME)~~
• ~~FROM STUDENT~~
• ~~SELECT LEN(ST_FNAME),~~
• ~~ST_FNAME~~
• ~~FROM STUDENT;~~
• ~~Select MAX(ST_FNAME)~~
• ~~FROM STUDENT~~
• ~~SELECT MAX(LEN(ST_FNAME))~~
• ~~FROM STUDENT~~
• ~~Select TOP(1) ST_FNAME~~
• ~~FROM STUDENT~~
order by LEN(ST_FNAME) DESC

• ~~SELECT POWER(SALARY, 2)~~
• ~~FROM INSTRUCTOR~~

• ~~Select CONVERT(VARCHAR(20),~~
~~GETDATE(), 101);~~

• ~~Select DB_NAME(), SUSERNAME()~~
ITI DESKTOP-VF50P25
• ~~DATABASE NAME(). ✓~~
• ~~SUSERNAME(). ✓~~

• ~~Create my own FUNCTIONS~~
• ~~String GetName(int id);~~

~~SQL~~ Engineer Adam

```
Create FUNCTION GetSname(@id int)
RETURNS VARCHAR(20)
BEGIN
    DECLARE @Name VARCHAR(20)
    SELECT @Name = st_fname
    FROM Student
    WHERE st_id = @id
    RETURN @Name
END
```

→ SELECT dbo.GetSname(3);
Mona AUNT.

. Get instructor total salary.

```
Create FUNCTION GetIst(@did int)
RETURNS TABLE
```

AS

RETURN

```
(SELECT ins_name, salary * 12 AS totalsal
FROM Instructor
WHERE DEPT_id = @did);
```

→ .SELECT * FROM GetIst(10);

ins_name totalsal

.Select ins_name FROM GetIst(10);
ins_name

SELECT SUM(totalsal) FROM GetIst(10);
→ 361152.00

~~SQL~~ ENGINEERADAM

Create FUNCTION Getsname(@Id int)
Returns varchar(20)

begin

declare @Name varchar(20)
SELECT @Name=st_fName
FROM Student
Where st_id=@Id
return @Name

end

→ SELECT dboGetsname(3);
Mona Aunty.

.Get instructor total salary.

create FUNCTION Getist(@did int)
Returns table

AS

return

(select ins_name, salary*12 AS totalsal
FROM INSTRUCTOR
Where DEPT_id = @did);

• SELECT * FROM Getist(10);
ins_name totalsal

Select ins_name FROM Getist(10);
ins_name

SELECT SUM(totalsal) FROM Getist(10);
361152.00

Create Function

```
setstmts1 @Format varchar(20)
returns @t table
(
    id int,
    ename varchar(20)
)
```

AS

begin

```
    if @Format = 'First'
        insert into @t
        SELECT st_id, st_fName
        From student
    else if @Format = 'Last'
        insert into @t
        select st_id, st_lName
        From student
    else if @Format = 'Full'
        insert into @t
        select id, FName + '' + LName
        From student
```

return

end

Concatenate string: FName + LName

اعرض الاسم

Select * From getstuds('Full');

SQL Queries.

Select * From grades

SELECT * FROM Grades;
 SELECT Sname, Grade,
 $X = \text{LAG}(Sname) \text{ OVER (ORDER BY Grade)}$,
 $Y = \text{LEAD}(Sname) \text{ OVER (ORDER BY Grade)}$,
 FROM Grades;

LAG & LEAD

SELECT Sname, Grade,
 $X = \text{LAG}(Sname) \text{ OVER (ORDER BY Grade)}$,
 $Y = \text{LEAD}(Sname) \text{ OVER (ORDER BY Grade)}$,
 FROM Grades
 WHERE SID = 4;

Sname	Grade	X	Y
1 Ahmed	90	NULL	Ahmed
2 Ahmed	97	Ahmed	Ahmed
3 Ahmed	100	Ahmed	Ahmed
4 Ahmed	110	Ahmed	NULL

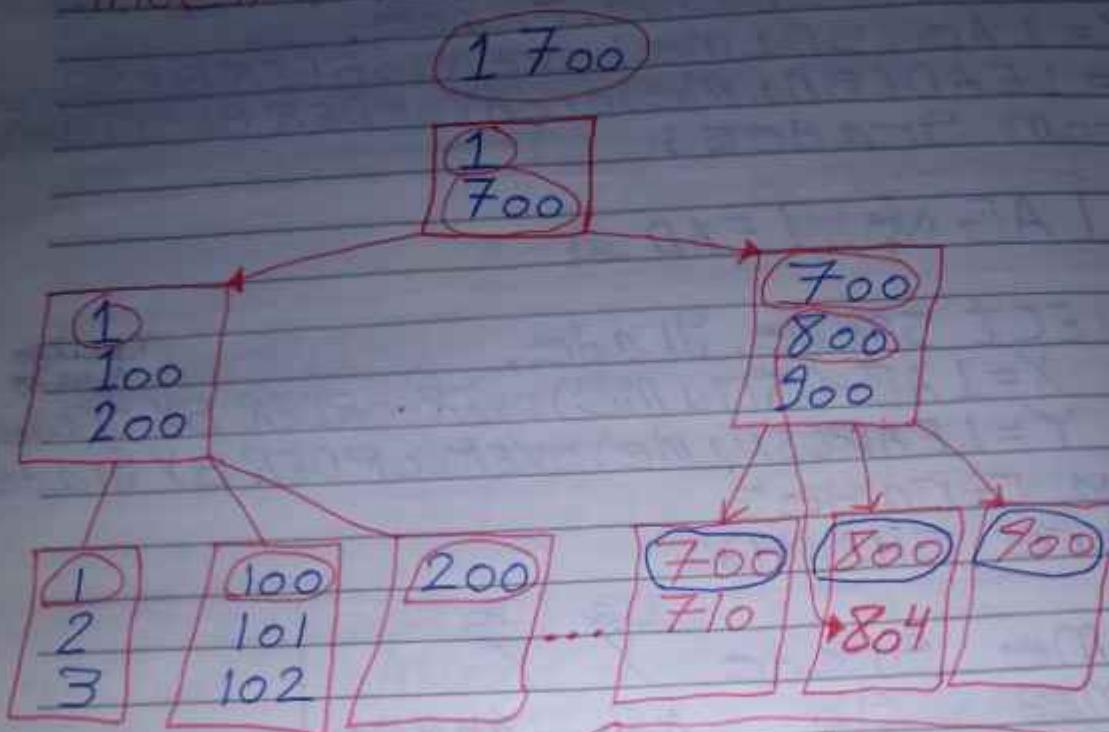
index. Heap ٩٥
 No Primary Key. Not sorted. غير مرتبة
 Table Scan. مسح الجدول
 مرجع

- PK
- Sorted
- Clustered Index

Select *
 From Student
 Where SID = 804

SID	Sname	age
1	AHMED	21
2	ALI	23
4	EMAN	24
7	OMAR	25

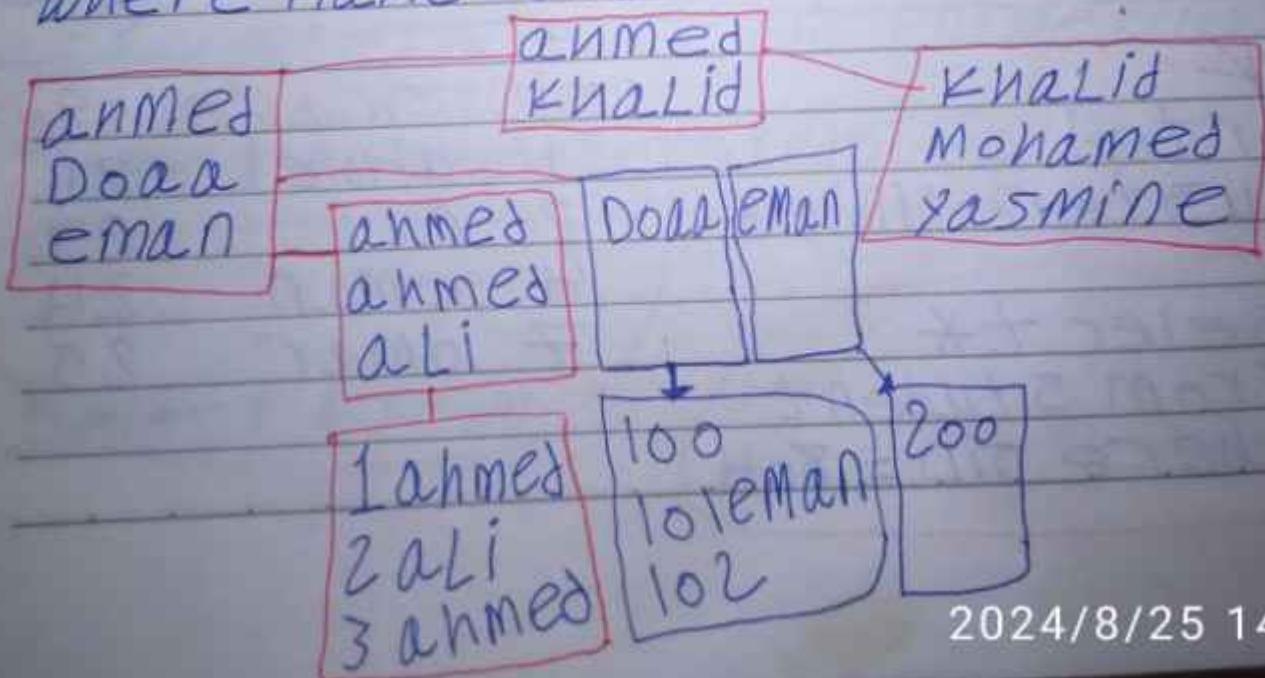
Index



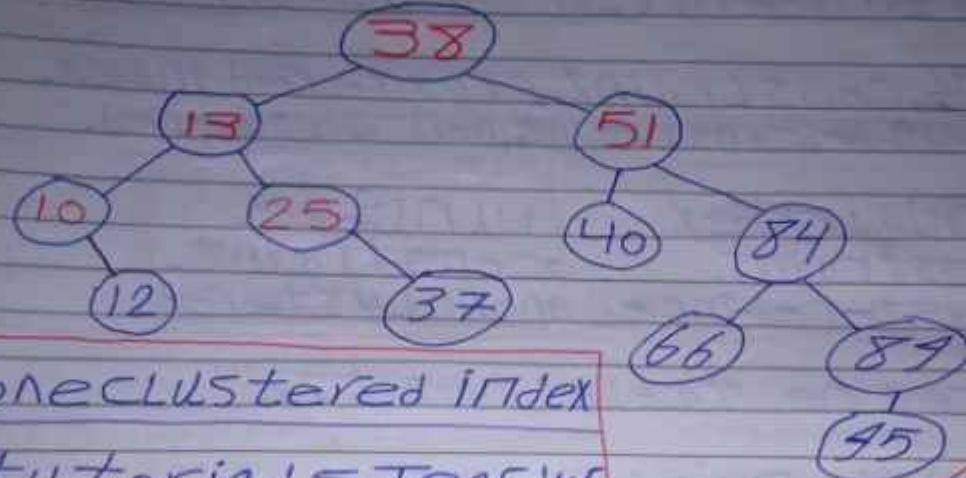
Binary tree. Select *
from student
where SID = 804

→ Nonclustered index(Sname).

select *
from student
where name = 'omar';



Binary Search Tree Example
38, 13, 51, 10, 12, 25, 40, 84, 25, 89, 37, 66, 95



- one clustered index
- tutorialsteacher

- SQL Server indexes:
clustered indexes.
Data in a table is stored in rows in an unordered structure called 'Heap'.
- When you create a primary key constraint on a table.
A unique clustered index is automatically created on the table.
- Create clustered index MYindex on Student(ST_Fname);
→ Cannot create more than one clustered index on table 'Student'.
one clustered.

→ Non clustered index
non clustered index
on student (st_id);

→ PK, constraint, clustered index.
unique constraint, non clustered.

→ PRIMARY KEY, UNIQUE
constraint, constraint.
→ clustered index, non clustered.

Create table test22

```
→ ID int PRIMARY KEY,  
    name varchar(20),  
    age int UNIQUE;  
Create unique index I4  
on student(st_age);
```

SQL Server Profiler.
SQL Server Audit.
SQL Server Tuning Advisor.
SQL Server Management Studio.

SEARCH: SQL Server Profiler.
SQL Server.
Trace Properties: RUN.

→ Database Engine Tuning Advisor
select database and tables to tune

→ SQL Server 2016 Database
Engine tuning Advisor.

CREATE nonclustered index
myIndex2
on STUDENT(ST_name);

PK constraint, clustered index.
unique constraint, non-clustered.

- | | |
|---------------------------|----------------------|
| <u>1. Primary Key.</u> | <u>unique.</u> |
| <u>2. Constraint.</u> | <u>constraint.</u> |
| <u>3. Clustered index</u> | <u>nonclustered.</u> |

create table test22

 id int primary key,
 name varchar(20),
 age int unique);
Create unique index i4
on STUDENT(ST_age);

SQL Server Profiler.

- . SQL Server Profiler +
- . SQL Server Tuning Advisor.
- . SQL Server Database Engine.

. Search :- SQL Server Profiler.

. SQL Server.

. trace Properties :- RUN.

→ Database Engine Tuning Advisor

→ Select Database and tables to tune

→ SQL Server 2016 Database
Engine tuning Advisor.

SELECT * FROM student
WHERE STAGE = 20;

Default Database :-

1. SQL Server DESKTOP.

2. Databases

3. System Databases

 MASTER

 MODEL

 TEMPDB

 TEMP TABLE

 VIEW

System Databases :-

→ All configuration of server.

→ Meta Data.

New Database. OK.

Create Table EXAM

(eid int,

 edate date,

 numOfQ int);

Drop Table EXAM;

Local Table → session based tables.

Global Table → shared table.

Create Table #EXAM

(EID int,

 edate date,

 numOfQ int);

Create table #exam
(
EID int,
EDate date,
numOfQ int);

Local table.

Create table #exam
(
EID int,
EDate date,
numOfQ int.);

Temporary Tables:-

dbo.#exam

dbo.#exam

Table Variable.

· Declare @t table(X int);
· Insert into @t values(1);
· Select * from @t;

use test;

Create table Sales
(

· PRODUCTID int,
· SALESMANNAME varchar(10),
· QUALITY int);

TRUNCATE TABLE Sales;

insert into SALES
values (1, 'Ahmed', 10)

(1, 'Khalid', 20)

(1, 'Ali', 45)

(2, 'Ahmed', 15)

(2, 'Khalid', 30)

COLUMN

• SELECT ProductID, SalesmanName, Quantity
From Sales;

Select ProductID as X, SUM(Quantity)
AS 'Quantities'
From Sales

Group By ProductID; X Quantities

↑ Union all

Select e.SUM(Quantity) | 1 210
From Sales; | 2 230

| 3 80

| 4 260

* this SQL ROLLUP option allows
the user to include extra rows
that represent the sub-totals.

SELECT ProductID as X,
SUM(Quantity) AS 'Quantities'
From Sales

Group By ROLLUP(ProductID); X 9

. 210 + 230 + 80 + 260 = 790. | 1 210

| 2 230

| 3 80

| 4 260

SELECT SalesmanName,
ProductID,

SUM(Quantity) AS 'Quantities'
From Sales

Group By CUBE(SalesmanName, ProductID)

→ SUM Every Group &
all row Sums

SELECT SalesmanName, ProductID,
 SUM(Quantity) AS Quantities
 FROM Sales
 GROUP BY Cube(SalesmanName,
 ProductID);

SalesmanName	ProductID	Quantities
ahmed	1	35
ali	1	145
Khalid	1	30
SUM NULL	1	210
ahmed	2	70
ali	2	90
Khalid	2	70
NULL	2	230
ahmed	3	60
Khalid	3	30
NULL	3	90
ali	4	170
Khalid	4	90
NULL	4	260
NULL	NULL	790 total
ahmed	NULL	165
ali	NULL	405
Khalid	NULL	220

EXAMPLE : For ID 1 = $35 + 145 + 30 = 210$

Select ProductID, SalesmanName,
 SUM(Quantity) AS Quantities
 From Sales
 Group By Grouping Sets (ProductID,
 SalesmanName)
 Order By SalesmanName

SELECT SalesmanName, ProductID,
 SUM(Quantity) AS 'Quantities'
 FROM Sales
 GROUP BY CUBE(SalesmanName,
 ProductID);

SalesmanName	ProductID	Quantities
ahmed	1	35
ali	1	145
Khalid	1	30
NULL	1	210
ahmed	2	70
ali	2	90
Khalid	2	70
NULL	2	230
ahmed	3	60
Khalid	3	30
NULL	3	90
ali	4	170
Khalid	4	90
NULL	4	260
NULL	NULL	790
ahmed	NULL	165
ali	NULL	405
Khalid	NULL	210

Total

EXAMPLE: For ID 1 = $35 + 145 + 30 = 210$

Select ProductID, SalesmanName,
 SUM(Quantity) AS 'Quantities'
 From Sales
 GROUP BY grouping sets(ProductID,
 SalesmanName)
 ORDER BY SalesmanName

SELECT GROUP BY PRODUCTID THEN SalesName.

, SELECT ProductID, SalesmanName,
SUM(Quantity) AS 'Quantities'
FROM Sales
GROUP BY Grouping Sets (ProductID,
SalesmanName)
order by SalesmanName;
→

ProductID	SalesmanName	Quantities
1	NVII	210
2	NVII	230
3	NVII	30
4	NVII	260
NULL	Ali	165
NULL	Alli	405
NULL	Khalid	220

. Pivot and UNPIVOT OLAP.
. If U have the result of the previous query.

Select ProductID, SalesmanName,
SUM(Quantity) AS 'Quantities'
From Sales
group by ProductID, SalesmanName;
→

ProductID	SalesmanName	Quantities
1	Ali	70
2	Ali	60
3	Ali	145
1	Khalid	90
2	Khalid	30
3	Khalid	70

• sets() group by ProductID Then SalesmanName.

• Select ProductID, SalesmanName,
SUM(Quantity) as 'Quantities'
From SALES
Group by grouping sets (ProductID,
SalesmanName)
order by SalesmanName;

ProductID	SalesmanName	Quantities	TotalSum
1	NVII	210	
2	NVII	230	
3	NVII	90	
4	NVII	260	
NULL	AHMED	165	
NULL	ALI	45	
NULL	KHALID	220	

• Pivot and UNPIVOT OLAP.

• If I have the result of the previous query.

• Select ProductID, SalesmanName,
SUM(Quantity) as 'Quantities'

From SALES

Group by ProductID, SalesmanName;

ProductID	SalesmanName	Quantities	SUM
1	AHMED	35	
2	AHMED	70	
3	AHMED	60	
1	ALI	145	
2	ALI	90	
4	ALI	170	
1	KHALID	30	
2	KHALID	70	
3	KHALID	30	

2024/8/25 14:56

Row becomes COLUMN.
→ GROUP BY COLUMNS.

PID	Ahmed	Ali	Khalid
1	35	145	30
2	70	90	70
3	---	---	---
4	---	---	---

SELECT * PI
FROM Sales
PIVOT(SUM(Quantity) FOR
SalesmanName IN ([Ahmed], [Khalid],
[Ali]) AS PVT

ProductID	Ahmed	Khalid	Ali
1	35	30	145
2	70	70	90
3	60	30	NULL
4	NULL	90	170

Group 1. Ahmed / 1. Khalid / 1. Ali
SUM .SUM .SUM

→ UNPIVOT opposite operation PIVOT,
rotating columns of table-valued.

→ Microsoft Learn (Search Google)
→ Using PIVOT and UNPIVOT-
SQL Server.

• FROM - Using PIVOT and UNPIVOT.

→ View.

- Is a **Select** statement.
- Specify user view of Data.
- Hide DB objects.
- Limit access of data. **Security**
- Construction of complex queries.
- Has no Parameter.
- Has no DML queries.
- View is virtual table.
- Only index view can increase Performance.

TYPES OF VIEWS:-

1. Standard view.
2. Partitioned View.
3. Indexed view.

Standard View:-

Create view VCAIRO
AS

```
SELECT ID, NAME, ADDRESS  
FROM STUDENT  
WHERE ADDRESS = 'CAIRO';
```

Create view VALEX
AS

```
SELECT ID, NAME  
FROM STUDENT  
WHERE ADDRESS = 'ALEX';
```

COLUMN



Create view VSTUDS

AS

```
Select *  
From MansServer.iti.dbo.STUDENTS  
Union All  
Select *  
From SonasServer.iti2.HR.STUDS
```

System views. dbo.VSTUDS.

Create view VSTUDS

AS

```
Select *  
From STUDENT;
```

Viewstudent. VSTUDS.

Select * From VSTUDS;

Create view VCairo

AS

```
Select id, FNAME, ADDRESS  
From STUDENT  
Where ADDRESS = 'Cairo';
```

→ Select * From VCairo;

→ Select FNAME From VCairo; ELSE NAME
• ALTER View VCairo (SID, SNAME, SADD)
AS

```
Select id, FNAME, ADDRESS  
From STUDENT
```

Where ADDRESS = 'Cairo';

• Select * From VCairo;

• Select SNAME From VCairo;

Create view Vall
AS

Select * FROM vCairo
UNION ALL

Select * FROM valEx

ALTER SCHEMA hr transfer Vall

→ SYSTEM VIEWS.

HR.Vall

We can make (inner join) in view.

SP_HELPTEXT 'VJoin';

INDEXED VIEW

(COPYDATA)

Create view vData

With SCHEMABINDING

AS

Select insName, salary
From dbo.Instructor
Where DeptId = 10;

Alter table instructor

Alter column insDegree varchar(50)

→ INDEX - TEMP TABLES - PIVOT - VIEW -
MERGE.

ALTER VIEW vCairo (SID, SName, SAdd)
AS
SELECT id, FName, Address
FROM STUDENT
Where ADDRESS = 'CAIRO'
With CHECK option;

٤٦ جبار الاختيار

2024/8/25 16:40

ALTER VIEW VEMPLOYEES

AS

```
SELECT *  
FROM Employees  
WHERE Title = 'SalesRepresentative';  
WITH CHECK OPTION;
```

GO;

Commands completed successfully.

MERGE:-

Merge into LASTTransaction
AS T ^{Subquery}
using(DAILY Transaction) AS S
ON T.ID = S.DID;

EX1

MERGE Locations T

using Locations Stage S ON

T.LocationID = S.LocationID

when MATCHED AND

T.LocationID = 3 THEN

DELETE

WHEN MATCHED AND

T.LocationID = 1 THEN

UPDATE SET

LocationName = S.LocationName;

Select * From Locations;

	LASTTRANSACTION			DAILYTRANSACTION		
	ID	name	myvalue	did	dname	dval
Taret	1	ahmed	4000	1	ahmed	4000
	2	ali	2000	2	ALI	1000
	3	omar	6000	10	Nada	3000
	4	eeman	7000			
	10	nada	3000			

- Stored Procedures & Triggers
- Stored Procedure, Triggers.
- Query → Parsing → optimize → (syntax) (metaData)
- Query tree → Execution Plan.
(Saved) From where Select (Memory).

Create Procedure Getst @id int AS

```
Select *
From Student
Where ID = @id; 4
```

Create Proc instst @id int,

AS @name varchar(20)

IF NOT EXISTS (SELECT ID

From Student

Where ID = @id)

.insert into student(id, name)

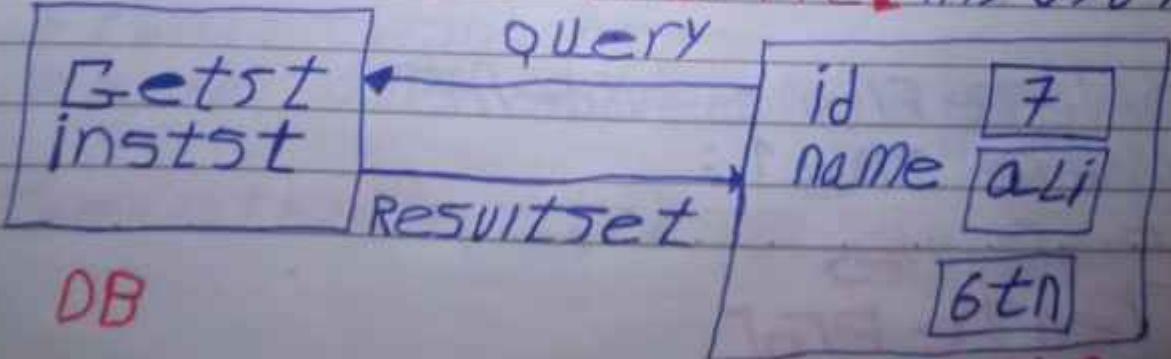
.values(@id, @name)

ELSE

Select 'DUPLICATE';

FIRSTCALL.

Network traffic ↗ instst ?, ali



Stored Procedures & Triggers

• Stored Procedure, Triggers.

• QUERY → PARSING → OPTIMIZE →
(SYNTAX) (METADATA)
→ QUERY TREE → EXECUTION PLAN.
(SAVED) FROM
WHERE
SELECT (MEMORY).

Create Procedure Getst @id int
AS

Select *
From Student
Where ID = @id;

Create Proc instst @id int.

@name varchar(20)

AS

IF NOT EXISTS (SELECT ID
From Student
Where ID = @ID)

. Insert into Student(id, name,
. values(@id, @name))

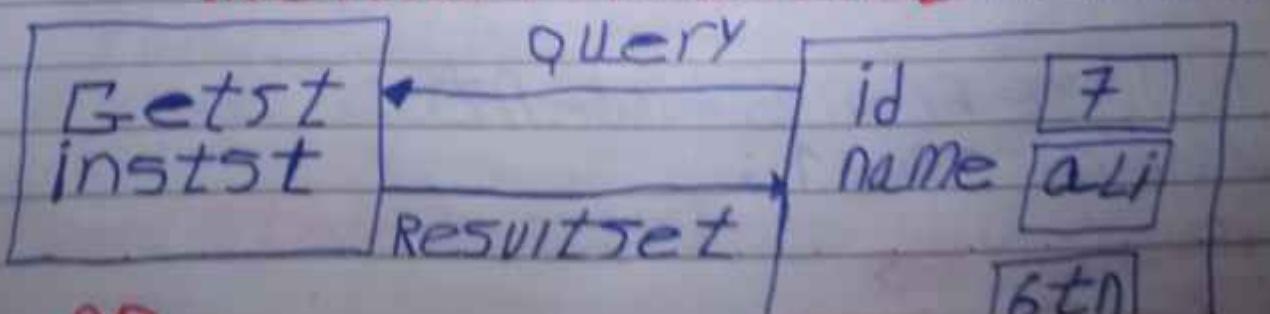
ELSE

Select 'DUPLICATE';

NoNameId

FIRSTCALL.

Network traffic ✓ instst ↴



DB

2024/8/25 16:42

APPLICATION

Connect to Server.

SQL Server

* SQL stored procedures take parameters.

* 3 types of SP.

* built-in SP.

- SP_bindrule.

- SP_unbindrule.

- SP_Def_constraint.

- SP_name.

- SP_duty.

* UserDefinedSP.

SELECT *

From student;

Create PROC Getst

AS

SELECT *

From student;

.Getst;

.Execute Getst;

.Create Proc GetStbyAddress @Add VARCHAR(20)

AS

Select ID, FNAME, address

From STUDENT

Where address = @Add;

.GetStbyAddress 'alex';

.Delete From student

Where ID = 1;

→ Procedures

.Create Proc.

```
create  
ALTER PROC Instst  
@id int, @name varchar(20)  
AS  
begin try  
    insert into student(id, fname)  
    values(@id, @name)  
end try  
begin catch  
    select 'Error'  
end catch
```

Instst 44, 'ali'

```
create Proc sumdata @X int, @Y int  
AS  
select @X + @Y int  
→ sumdata 3, 9 - Calling Parameter by Position.
```

Calling Parameter by Position.

sumdata @Y = 9, @X = 4 Calling Parameter

.sumdata 3

```
create PROC GetstbyAge @age1 int,  
@age2 int
```

AS

```
SELECT ID, fname  
From Student  
Where age between @age1 And  
@age2
```

GetstbyAge '23, 28'

insert into tab4(ID, Fname)
execute GetStbyAge 23, 28

Declare @t table(X int, Y varchar(20))
insert into @t
execute GetStbyAge 23, 28
Select count(*) From @t
→ 4

ALTER PROC GetData @ID int,
@age int OUTPUT,
@name varchar(20) OUTPUT

A5

SELECT @age = st.age,
@name = st.fname
From student
Where st.ID = @ID;
DECLARE @X int, @Y varchar(20)
execute GetData 3,
@X OUTPUT,
@Y OUTPUT
select @X, @Y
27, ADAMS

trigger.

1. can't call.
2. can't send parameter.
3. triggers on table.
4. insert, update, delete.

* Triggers is constraint.
* A constraint is condition.

Create trigger t1
on STUDENT
AFTER INSERT
AS

Select 'Welcome to ITI';

Create trigger t2
on STUDENT
FOR UPDATE
AS

Select getDate();

UPDATE STUDENT
SET STAGE += 1
→ 2020-11-29 12:19:18.993

Create trigger t3
ON STUDENT
INSTEAD OF DELETE
AS

Select 'Not allowed for user = '+username()

Delete From STUDENT
Where ID = 779;

→ 'Not allowed for user = DESToP-VF5oP25.'

Create trigger t4
ON DEPARTMENT
INSTEAD OF INSERT, UPDATE, DELETE
AS

Select 'Not allowed'

~~UPDATE Department
set name = 'Cloud'
where Dept ID = 40;
.1 Not allowed~~

~~ALTER trigger SALES.t7
on SALES.student
AFTER UPDATE~~

~~AS
if UPDATE(name)
select 'hi'~~

~~UPDATE SALES.student
set name = 'Ahmed'
where ID = 7;~~

~~DROP trigger t4;~~

~~ALTER TABLE Department
DISABLE trigger t4;
ALTER TABLE Department
ENABLE trigger t4;~~

~~Select * From inserted~~

~~Select * From deleted~~

~~Create table History
(_User varchar(20),
_date date,
_oldid int,
_NewID int);~~

instead of update
as

```
IF UPDATE(toPID)
begin
DECLARE @new int, @old int
SELECT @old = toPID FROM Deleted
SELECT @new = toPID FROM inserted
INSERT INTO History
VALUES(SUSERNAME(), GETDATE(),
@old, @new)
END
```

OUTPUT.

```
Delete From Student
OUTPUT getDate(), Deleted.ST_Fname
Where ST_id = 44
```

2020-11-29 13:24:04.280 | ali

```
insert into student(ID, Fname)
OUTPUT 'welcome to iti'
VALUES(444, 'ali')
```

XML

For XML ::

1. For XML raw.
2. For XML auto.
3. For XML EXPLICIT.
4. For XML PATH.

OPEN XML

SELECT *

From STUDENT

FOR XML raw; convert to XML

2024/8/25 16:46

```
SELECT *
FROM student
FOR XML raw('student');

SELECT *
FROM student
FOR XML raw('student'), ELEMENTS
<student>
  <st_ids>5<list_ids>
    <st_lnames>Mohamed<list_lnames>
      <st_ages>30<list_ages>
        <st_address>Alex<list_address>
          <st_ages>30<list_ages>
            <dept_ids>10<list_dept_ids>
              <st_supers>1<list_stsupers>
                <list_students>
```

```
Select *
From student
For XML raw('student'),
ELEMENTS,
Root('ITI_Studs')
```

→ How to show null values in XML.

```
Select * From student
For XML raw('student'),
ELEMENTS XSiNil,
Root('STUDENTS');
```

```
Select Address, Count(ID) From student
Group By Address
For XML raw('student'), ELEMENTS,
Root('STUDENTS');
```

<STUDENTS><student>6</student>

- U CAN ONLY PRESENT DATA AS ELEMENTS OR ATTRIBUTES.

JOIN PROBLEM.

```
SELECT topic.TopID, TopName,  
       crs.ID, crsName  
  FROM Topic, Course  
 WHERE Topic.TopID = Course.TopID  
 FOR XML AUTO, ELEMENTS;
```

PATH.

```
select ID 'studentID',  
      Fname 'studentName/FirstName',  
      Lname 'studentName/LastName',  
      Address 'Address'  
  FROM Student  
FOR XML PATH('student');
```

```
<student studentID='1'>  
  <studentName FirstName='ali'>  
    <lastName>HASSAN</lastName>  
  <studentName>  
    <Address>CAIRO</Address>  
</student>
```

```
FOR XML PATH('student')
```

- MICROSOFT FOR XML (SQL SERVER).
- RAW AUTO EXPLICIT PATH.
- FOR XML (SQL SERVER). Microsoft (learn)

1. Create Proc PROCESS tree.

DECLARE @hdocs XML = 'XMLCode'

2. Declare DOCUMENT handle

Declare @hdocs int

3. Create Memory tree

EXEC SP XML_Preparedocument

@hdocs output, @docs

4. Process Document read tree

From memory.

5. OPENXML Create Resultset
From XML Document.

SELECT * into ITISTUDS

FROM OPENXML(@hdocs, 'liststudent')

WITH (ID int '@studentID',

 Addr varchar(10) 'Address',

 SFirst varchar(10)

 'studentName/Second')

 WITH (ID int '@studentID',

 Addr varchar(10) 'Address',

 SFirst varchar(10)

 'studentName/First',

 SSECOND varchar(10)

 'studentName/Second'

)

5) Remove Memory tree.

EXEC SP XML_RemoveDocument (@hdocs)

Select * From ITISTUDS;

Select * From ITISTUDS;

Microsoft Learn: XMLdata (sqlserver).

1. FOR XML (SQLServer).

2. OPENXML (transact-SQL).

2024/8/25 16:53

- CURSOR
- HOW to DEAL RESULTSET ROW BY ROW.
- FOR LOOP FOR ROWS.

```
Select ID, Fname  
From STUDENT  
Where ADDRESS = 'Cairo';
```

```
• DECLARE C1 CURSOR  
For Select ID, Fname  
From STUDENT  
Where ADDRESS = 'cairo'  
For read ONLY --UPDATE.
```

```
• DECLARE @ID int, @Name Varchar(20)
```

```
Open C1
```

```
Fetch C1 into @id, @name
```

```
While @@FETCH_STATUS = 0
```

```
begin
```

```
    select @id, @name
```

```
    Fetch C1 into @id, @name
```

```
end
```

```
close C1
```

```
deallocate C1
```

```
DECLARE C1 CURSOR  
FOR SELECT Fname  
From STUDENT  
where Fname is not NULL  
FOR READ ONLY  
DECLARE @Name VARCHAR(20),  
@AllNames VARCHAR(300) = ''  
OPEN C1  
FETCH C1 INTO @Name  
WHILE @@FETCH_STATUS = 0  
begin  
    SET @AllNames = Concat(  
        @AllNames, ',', @Name)  
    FETCH C1 INTO @Name  
end  
SELECT @AllNames  
CLOSE C1  
DEALLOCATE C1
```

- MICROSOFT LEARN.
- CURSORS (SQL SERVER). ✓
 - 1. DECLARE CURSOR.

the Fetch-Execute.

جلب - إفادة

TYPES OF BACKUPS.

- FULL BACKUP.
- DIFFERENTIAL BACKUP.
- TRANSACTION LOG BACKUP.
 .MDF
- FILEGROUP BACKUP .IDF

• Right Click Database.

→ New Database. OK.

right click → Tasks → BACKUP

→ ITI Database → Tasks → BACKUP

• BACKUP Database - mytest

* BACKUP TYPE: FULL.

* Locate Database Files.

* mytest.bak → OK.

* OK.

• Microsoft Learn. أمثلة فحص

. BACKUP overview (SQL Server).

. Best BACKUP MODE?

→ A FULL BACKUP is most complete type of backup. ~~it's~~ restore data.

• SQL Server 13.0.1742.0 → Database.

. Restore Database.

. Restore Database -.

. Locate BACKUP FILE.

. BACKUP timeline: mytest. OK.

• BACKUP Database ITI

to DISK = 'd:\iti_db.bak'

BACKUP Database ITI

to DISK = 'd:\iti_db.bak';

. SQL Server Agent

. jobs

. Make jobs - starts jobs.

- . types of insert statement.
- . simple insert.
- . insert constructor.
- . insert based on select.
- . insert based on execute.
- . bulk insert.

. bulk insert emp
From 'd:\mydata.txt'
with(Fieldterminator=',')

. SQL Server Import
And Export Wizard.

BI. Microsoft Access.

- . Snapshot DB.
- . 80% empty.
- . Read Only DB.
- . Snapshot DB (30/11/2000) 11 am
- . iti

SID	SNAME	AGE
1	Ahmed	22
2	Eman	21
3	Omar	25
4	Sally	22

Copy & Write

SELECT *
From student;

Microsoft SQL Server
Management Studio.

GivenName.ss
DatabaseProperties-ITI

```
CREATE DATABASE itisnap
ON
(
    NAME = 'ITI',
    FILENAME = 'd:\iti\itisnap.ss'
)
AS SNAPSHOT OF iti
```

Select * From student;

```
RESTORE DATABASE ITI
FROM Database_SNAPSHOT='ITISNAP';
*SNAPSHOT NAME in object explorer.
```

* CURSOR. CURSOR.
* BACKUP. BACKUP.
* SNAPSHOT. SNAPSHOT.
* SQLCLR. SQLCLR.

SQLCLR

Function

NewData type [CLASS STRUCT]

```
SP_CONFIGURE 'CLREnable', 1
GO
```

RECONFIGURE

'CLREnable' changed from 1 to 1.

. Give Name.SS

. Database Properties - ITI

. Create Database itisnap

ON

(

Name = 'ITI',

FILENAME = 'D:\ITIISNAP.SS'

)

AS SNAPSHOT OF ITI

. Select * From student;

. Restore Database ITI

. From Database SNAPSHOT = 'ITISNAP';

* SNAPSHOT NAME IN OBJECT EXPLORER.

* CURSOR. CURSOR.

* BACKUP. BACKUP.

* SNAPSHOT. SNAPSHOT.

* SQLCLR. SQLCLR.

SQLCLR

Function

New Data type [CLASS STRUCT]

. SP Configure 'clr enabled', 1

GO

RECONFIGURE

'clr enabled' changed from 0 to 1.

*MICROSOFT LEARN
INTRODUCTION TO SQL SERVER
CLR INTEGRATION.

• SQL CLR
• ENABLING CLR INTEGRATION.

SQl

• SP_CONFIGURE 'CLRENABLED', 1

GO

RECONFIGURE

GO

• VISUAL STUDIO CODE.

FILE → NEW → PROJECT.

SQL SERVER → OK → DATABASE12

→ ADD → EXISTING ITEM >>

SQL CLR C# >> SQLCLR C#

USER DEFINED FUNCTION.

• PUBLIC PARTIAL CLASS.

• USING SYSTEM.DATA.SQLTYPES;

• PUBLIC PARTIAL CLASS

USERDEFINEDFUNCTIONS{

[MICROSOFT.SQLSERVER.SERVER.
SQLFUNCTION]

PUBLIC STATIC SQLINT16

SUM2INT(SQLINT(SQINT16 X,
SQINT16 Y))

{

//PUT YOUR CODE HERE.

RETURN X+Y;

} }

~~SP_configure 'clr enable', 1~~
So
~~reConfigure~~
• ~~SELECT dbo.SUM2INT(3,5)~~
• ~~SELECT dbo.SUM2INT(DeptID,Dept)~~
Select dbo.SUM2INT(ID,age)
From STUDENT;

TOPICS SQL :-

1. crystal reports
2. SQL INJECTION
3. Red-Gate (SQL Development Tools Only)
4. Data Quality Services
5. Power View (Shimaa)
6. Power Query
7. Power Pivot
8. Target BI
9. What is new in SQL Server 2019 as Development
10. DB Mirroring
11. DB ENCRYPTION
12. SMO
13. SQL Server Snapshot
14. Hadoop & MAP Reduce and Hive.

TUTORIALS Teacher:

SQL Server INDEXES: Clustered INDEXES.

لندن

في الاتصالات و تطبيقات BI.

BI

في حالك ملحوظة.

MSBI Lectures:

Business intelligence
Software.

MSBI.

ما هو MSBI وما هي MSBI

Microsoft Business Intelligence
which consists of tools that
help in providing enhanced
solutions to business intelligence
and data mining queries.

MSBI uses a SQL Server and
Visual Studio Platform that
enables users to better access
and information.

Business and Data Mining Queries
الاستعلامات الاعمال واستخراج البيانات

1. SSIS.

2. SSAS.

3. SSRS.

* MSBI - Microsoft Business
Intelligence.

1. SSIS - SQL Server integration
services.

2. SSAS - SQL Server integration
services.

3.

* M5BI - Microsoft Business Intelligence:

1. SSRS - SQL Server Reporting Services.
2. SSIS - SQL Server Integration Services.
3. SSAS - SQL Server Analysis Services.

* SSIS - SQL Server Integration Services. MOST important.

* BI - DAY 1

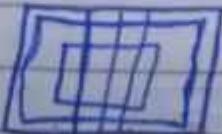
1. Market Analysis. تحليل السوق.
2. Data Quality. جودة البيانات.
3. Integration. تكامل رفع.
4. Reports. التقارير.
5. Dashboard. لوحة.
6. Very Large Data. بيانات كبيرة جداً.
7. Historical. تاريخي.
8. Data Mining.
9. ETL.
10. Data Warehouse. مستودع البيانات.

→ Data Mining: analyzing large databases to generate new information.

مصدر البيانات.

Data Source.

FILE



EXCEL

2024/8/25 17:04

On 180 MBA

- DB.
- Oracle, SQL, MySQL.
- XML
- Binary Files.

12 MIN

• Datasource جهاز
البيانات

- Meta Data.
- LDF MDF.
- Integration چیزی داده ای که در دو دیتابیس میباشد.
- ETL Tools.

→ Extract transform Load.

* Dataware House. MAKE ANALYSIS.

DB.

* Denormalized DB. Less Table.

Historical DB.

* insert-update-delete-select.

want no problem in this queries.

→ relational DataBase NO TRUST.

* star schema.

BUILD Dataware House :-

VIP 1. MEASURES [AGG VALUES]; ✓

VIP 2. PERFORMANCE [SELECT]; ✓



جامعة مدارس MBA

12 MIN

- DB.
- ORACLE, SQL, MySQL.

L1

L2

L3

L4

- XML
- BINARY FILES.

- Metadata.

- LDF MDF.

- Integration ETL. Datasource.

- ETL Tools.

Extract transform Load.

* Dataware House. MAKE ANALYSIS.

DB.

* Denormalized DB. Less table.

Historical DB.

* insert-update-delete-select.

→ want no problem in this queries.

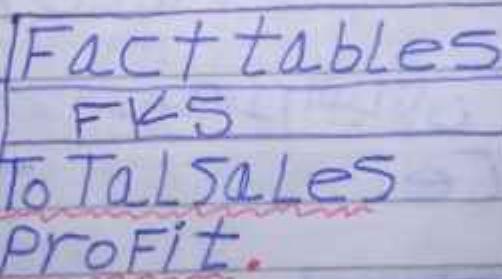
→ relational DataBase NOT TRUST.

* star schema.

Build Dataware House :-

VIP 1. MEASURES [AGGVALUES]; ✓

VIP 2. PERFORMANCE [SELECT]; ✓



product

branch

EMPLOYEE

CUSTOMERS

* steal ideas from others.

الرهان والقرار حام.

- DATABASE.
- E-DATAwareHouseDBIV. ① بيانات ايجابية
- HISTORICAL DB.
- MAKE ANALYSIS (CUBES), Data Mining.
- MAKE Reports - Select.
- DATA Mining :- GROUP OF MATHEMATICAL ALGORITHM APPLIED ON DATA. To MIN
- Data Mining USES :-
 1. PREDICTION.
 2. CLASSIFICATION.
 3. ANALYSIS.

* BI :- → STEPS OF BI :-

1. Integration.
2. Analysis.
3. Reports.

→ BI steps :-

1. SQL Server Integration Services.
2. SQL Server ANALYSIS SERVICES.
3. SQL Server Reporting Services.

- MARKET ANALYSIS.
- Data QUALITY.
- Integration.
- Reports.
- Dashboard.
- Very Large DATA.
- historical
- Data Mining.
- ETL....
- DataWarehouse.

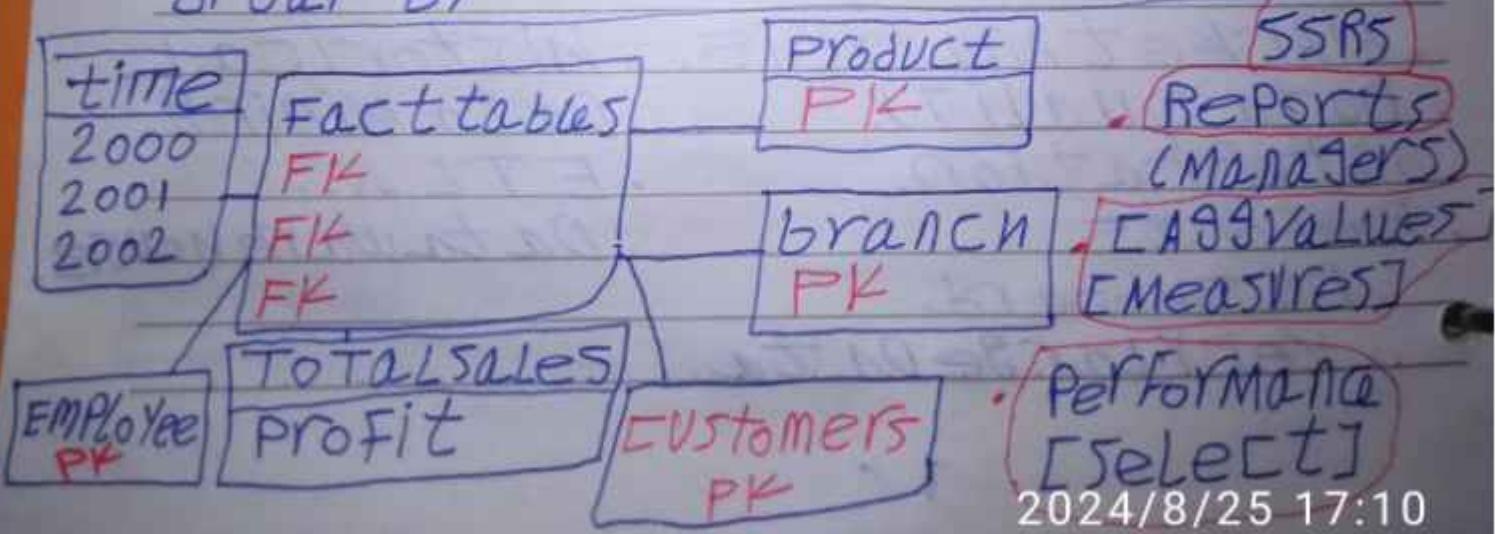
DATA INTEGRATION

DATA SOURCE:-



STAR SCHEMA:-

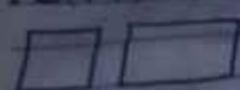
```
Select SUM(TotalSales), Pid, Bid
FROM ...., ...., ....
    Joins
GROUP BY
```



2024/8/25 17:10

iamAddMS

DATA SOURCE :-



FILES

ETL tools

Extract - Transform - Load



Integration

Packages

EXCEL

DB

ORACLE

SQL

MySQL

XML

XML

Binary Files.

DATAWAREHOUSE

DB

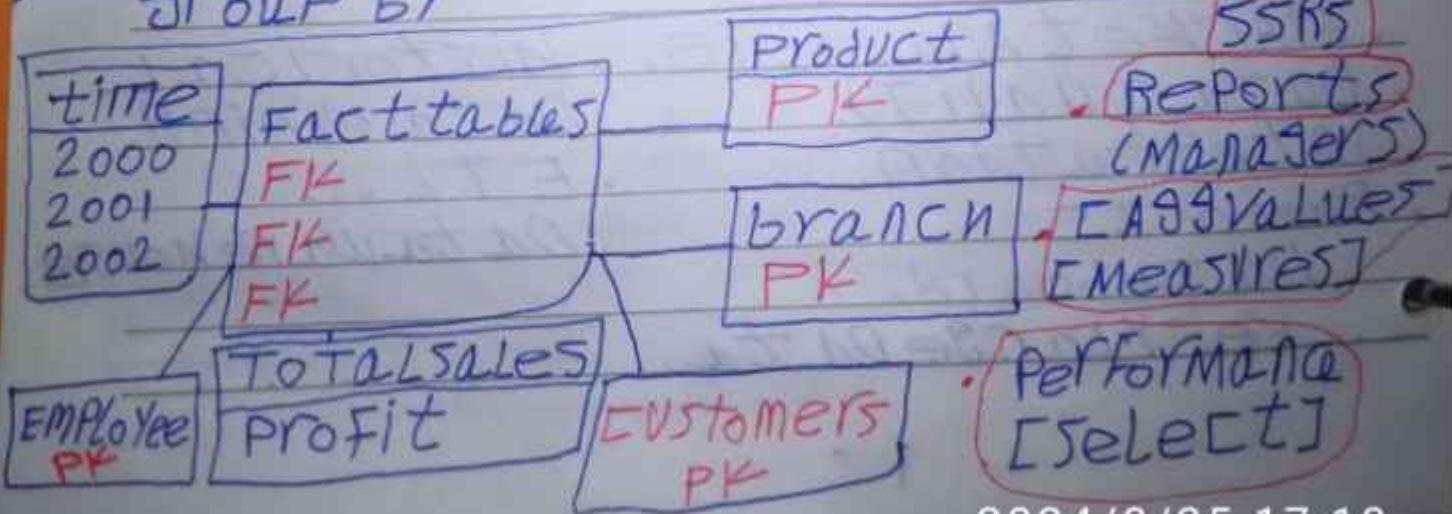
DeNormalized DB

HISTORICAL DB

SSIS

STAR SCHEMA :-

Select SUM(TOTALSALES), Pid, Bid
From.....,.....
Joins
Group By



- Integration SSIS.
- Analysis cubes. SSAS. Data mining.
- Reports (Managers). SSRS.

• Microsoft Business intelligence.

• What is Microsoft Business intelligence:-

1. Integration. Datacamp

2. Stores AND ANALYSIS.

3. Dashboards AND REPORTS.

HELP TO MAKE BETTER BUSINESS DECISIONS.

BUSINESS INTELLIGENCE GROUP OF APPLICATIONS AND TOOLS.

* What is Business Intelligence?

Analyzing useful information from the data in an organization.

* Getting the right information, to the right decision makers.

* Forecast future results.

* BI LEADS TO:

1. Fact-based Decision Making.

2. Single Version of Truth.

* BI INCLUDES Reporting and Analytics.

• Data warehouse.

→ Must Truth.

MICROSOFT BUSINESS INTELLIGENCE:-

* MICROSOFT SQL SERVER
INTEGRATION SERVICES.

INTEGRATE

- Data from source systems & integration.
- Data transformation.
- ANALYZE
- Historical data.
- Business logic.
- Hierarchical views.
- Data discovery via data mining.

REPORT

- Data presentation & distribution.
- Data access.

• MICROSOFT SQL SERVER
INTEGRATION SERVICES.

• MICROSOFT SQL SERVER
ANALYSIS SERVICES.

• MICROSOFT SQL SERVER
REPORTING SERVICES.

• MICROSOFT BI PLATFORM.

• DELIVERY.

• END USER TOOLS AND

PERFORMANCE MANAGEMENT.

• BUSINESS INTELLIGENCE PLATFORM.

2024/8/25 17:13

الطب البشري

MICROSOFT BUSINESS INTELLIGENCE:-

* MICROSOFT SQL SERVER
INTEGRATION SERVICES.

Integrate

- Data from source systems & integration.
- Data transformation.
- ANALYZE
- Historical Data.
- BUSINESS LOGIC.
- Hierarchical views.
- Data discovery via Data mining.

Report

- Data presentation & distribution.
- Data ACCESS.

• MICROSOFT SQL SERVER
INTEGRATION SERVICES.

• MICROSOFT SQL SERVER
ANALYSIS SERVICES.

• MICROSOFT SQL SERVER
REPORTING SERVICES.

• MICROSOFT BI PLATFORM.

• DELIVERY.

• ENDUSER TOOLS AND
PERFORMANCE MANAGEMENT.

• BUSINESS INTELLIGENCE

IMPLEMENTATION OF BI

* Designing a BI Application involves multiple layers.

1. Data Sources. * . META DATA
2. Data Integration. * . PERFORMANCE
3. Data Storage. * . OPERATIONS
4. Data Analysis. * . SECURITY
5. Data Presentation. *

. OLTP Database.

. OLAP Database.

. DATAWAREHOUSE. DATABASE.

Data Warehouse(OLAP) AND OLTP

. Datawarehouse

. OLTP

. wide information.

. small pieces information.

DATA

. Update on a schedule.

. Updated in real-time.

LESS TABLES

. DeNormalised

Normalised

. Large Very Large Database.

. Small to Large Database.

→ Read Queries.

Update, insert queries.

. Non-Volatile.

Volatile.

. APPLICATIONS that
that ANALYSE
the BUSINESS.

APPLICATION that
run the BUSINESS.

الخط
What is CUBE?

1. MEASURES.
2. DIMENSIONS.
3. AGGREGATIONS.



. REPORTS FROM CUBE? 1:26

DATA PRESENTATION.

→ 1. USING EXCEL TABLES.

2. USING SQL SERVER

REPORTING SERVICES SSRS.

→ 3. UNDERSTANDING SHAREPOINT.

4. REPORTING TOOLS.

5. 3RD PARTY TOOLS.

BI → DATA ENGINEER → DATA SCIENCE
→ MACHINE LEARNING ENGINEER.

1:53

BI_DAY_1

مختص في هذا الأسبوع

BIG DATA ENGINEER. توظيفات . حق .

1. PYTHON.

2. DATABASE AND SQL. SQL SERVER.

3. ETL AND DATA WAREHOUSING.

SSIS:-

• USED TO GENERATE PACKAGES.

• CONTROL FLOW.

• DATA FLOW.

• PRECEDENCE CONSTRAINTS.

• CONNECTION MANAGERS.

• VARIABLES.

• DATA VIEWERS.

control flow

CONTROLFLOW: MAKE TASKS.



DataFlow.

onSuccess.

onFailure.

onComplete.

.sort

.oracle

connection Manager.....

step BY step: SEARCH.

1. SQL Server Data Tools 2015.

2. File → New → Project.

New Project :-

→ TEMPLATES.

→ BUSINESS intelligence.

* ANALYSIS SERVICES.

* INTEGRATION SERVICES.

* REPORTING SERVICES.

ControlFlow:- EXAMPLE.5.

1. EXECUTE SQL TASK.

2. FILESYSTEM TASK.

3. FOREACH LOOP CONTAINER.

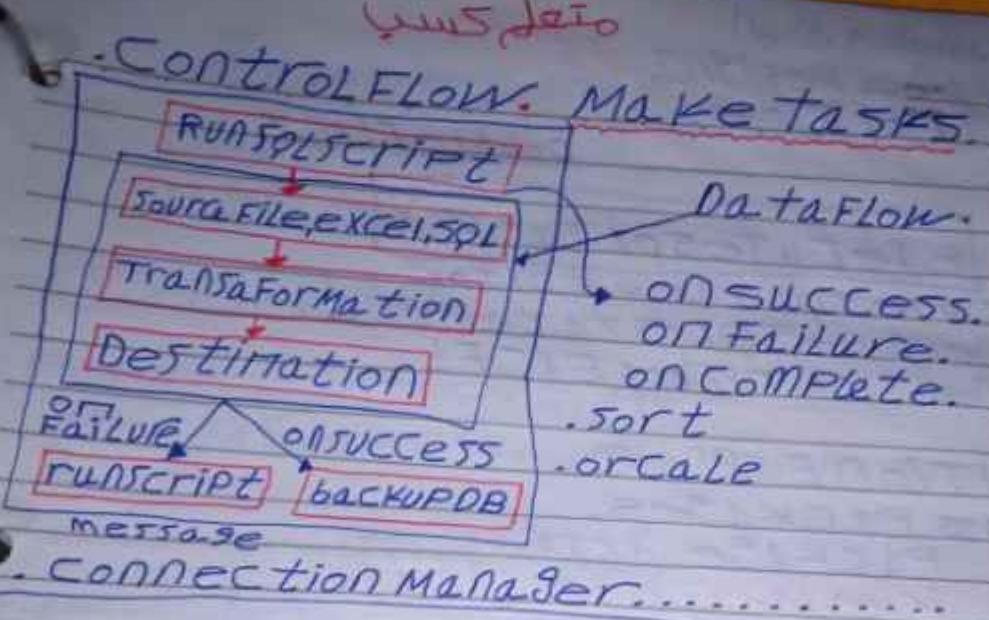
4. FOR LOOP CONTAINER.

5. BULK INSERT TASK.

6. SCRIPT TASK.

7. BACKUP DATABASE TASK.

8. SHRINKDATABASE TASK.



Step by Step: Search.

1. SQL Server Data Tools 2015.
2. File → New → Project.

New Project:-

- TEMPLATES.
- BUSINESS INTELLIGENCE.
 - * ANALYSIS SERVICES.
 - * INTEGRATION SERVICES.
 - * REPORTING SERVICES.

Control Flow:- Examples.

1. EXECUTE SQL TASK.
 2. FILE SYSTEM TASK.
 3. FOREACH LOOP CONTAINER.
 4. FOR LOOP CONTAINER.
 5. BULK INSERT TASK.
 6. SCRIPT TASK.
 7. BACKUP DATABASE TASK.
 8. SHRINK DATABASE TASK.
- 2024/8/25 17:18

الحال و البنون ~~الأفضل~~

→ PACKAGE[DESIGN].

CONTROLE FLOW

DOUBLE CLICK

EXECUTE SQL TASK X

EXECUTE SQL TASK EDITOR

General → SQL Statement.

→ CONNECTIONTYPE: EXCEL.

SOLUTION EXPLORE.

SSIS Packages.

PACKAGE.dtsx

View Code [XML].

SQL Statement.

CONNECTIONTYPE: OLE DB.

CONNECTIONTYPE: OLEDB.

CONNECTION: DATA CONNECTIONS:

Server Name: -

Connect to Database:

→ LOCALHOST ITI → Select: ITI

SQLSourceType: Direct Input

SQLStatement:

→ Insert into student
(st_id, st_fName)
VALUES(100, 'omar');

BACKUP DATABASE TASK X

CONNECTION: New...

CONNECTION NAME:

Con1

BACKUP: TYPE: - FULL

DATABASE: ITI

DESTINATION FOLDER: F.

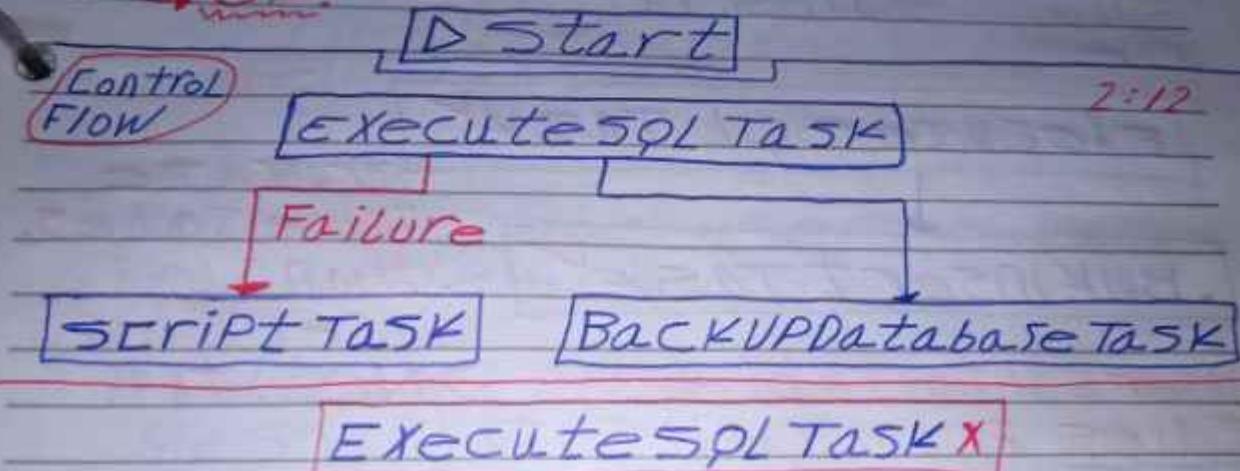
2024/8/25 17:18

SCRIPT TASK EDITOR

VScript.

- Script Language: Microsoft Visual C# 2015.
- → **EDIT SCRIPT**

```
Public Void Main()
    MessageBox.Show("error in Insert");
    } → OK
```



SQl Statement

Connection Type: OLE DB

Connection: Localhost\ITI

SQLSourceType: Direct Input

SQLStatement:

truncate table SALES;

SALES.TXT BULK INSERT TASK COPY PAST FILE

• Connection. → **BULK INSERT TASK EDITOR**

• A Source Connection.

• File : D:\SALES.TXT **BROWSE**

• OK.

• Format.

COLUMN Delimiter: [comma],

ROW Delimiter: [CR][LF]

محتوى الورقة

- Open Bulk Insert Task Editor
- Connection :-
 - Connection LocalHost\ITI
 - Destination Table [dbo].[SALES]
- Format
 - Format Specify
 - Row Delimiter [CRLF]
 - Column Delimiter [comma], ,
- Source Connection
 - File SALES.txt
 - OK

- Execute SQL Task ✓ 2:18
- ↓
- Bulk Insert Task ✓
 - TRUNCATE TABLE SALES.
 - Then
 - INSERT INTO SALES FROM SALES.txt
- Insert into Table SALES FROM SALES.txt

For Loop Container ①

→ Script Task

Variables

Name	Scope	Datatype	Value
Counter	PACKAGE2	int32	②

Double Click → For Loop Editor.

For Loop EDITOR

✓ For Loop Properties

• InitExpression @Counter = 1

• EvaluateExpression @Counter = 5

• AssignExpression @Counter = @Counter + 1

✓ General

Name ForLoopContainer

Description ForLoop Container.

OK

SCRIPT TASK EDITOR

• ScriptLanguage MICROSOFT VISUAL
C# 2015.

• EntryPoint Main

• ReadOnlyVariable User::Counter

•ReadWriteVariables

→ EDIT SCRIPT....

```
PUBLIC Void Main()
    MessageBox.Show(
        DTS.Variables["Counter"].  

        Value.ToString());
```

ControlFlow ▷ START

ForLoopContainer ✓

↳ Script Task

MetroSales:-

Alex

→ We Need For Loop.

Cairo

MANSOURA

عربي مصر

SALES.txt

New Package

Control Flow

Foreach Loop Container

- Collection:-
- Foreach Loop Editor
- Enumerator: Foreach File Enumerator
- Enumerator Configuration
- Folder:
F:\Metrosales
- Files:
*.txt
- Retrieve FileName
 - Fully Qualified
- ✓ Traverse SubFolders.

Variable Mapping

Variable INDEX

Add Variable

Name: MYPATH

OK

User::MYPATH

OK

Foreach Loop Container

SCRIPT TASK

- Script Language: Microsoft Visual C# 2015
- EntryPoint: Main
- ReadOnlyVariables User::MYPATH
- EDIT SCRIPT....

DTS Data transformation services.

\Project

Public void Main() {
 MessageBox.Show(Dts.Variables["
 "MYPath"].Value.ToString()); }

Start.

→ F:\Metrosales\SALES.txt

foreach loop container

script task ✓

Double Click DataFlow A.

The DataFlow SQL UI lets you
create SQL queries to run DataFlow
jobs.

* MULTICAST: EXCEPT 3 types.

→ COPY FILE TO XML, FILE, EXCEL.

* Conditional Split: ^{join} Conditional Split.
→ Copy according to condition.

* Union All: EXCELSHEET, TEXTFILE.

Merge Join: JOIN, LEFT OUTER JOIN,
INNER JOIN, RIGHT OUTER JOIN.

Merge Join: Merge inputs must
be sorted.

• MULTICAST.

• CONDITIONALSPLIT.

• UNION ALL.

• MERGE JOIN.

Merge.

• ExpressionBuilder
• variables and Parameters
• System Variable
 >User::MyPath

EXPRESSION:

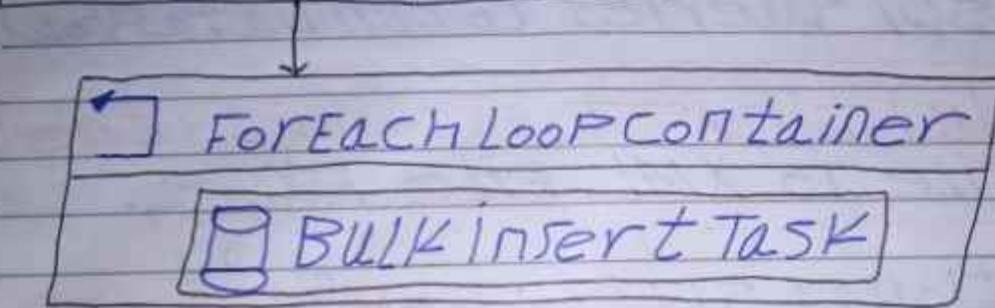
(@User::MyPath) Dynamic Variable

OK

connection Managers:-

Sales.txt LocalHost.1TI)

Execute SQL Task



DataFlow Double Click

Source Assistant

Select Source Type:-

• SQL Server

Connection Manager

• EXCEL

Server Name:

• FLATFILE

• connect to Database

• ORACLE

• select or enter Database:

→ ITI

OK

OK

2024/8/25 17:24

→ OLE DB SOURCE EDITOR.

→ OLE DB CONNECTION MANAGER:
LOCALHOST. ITI

→ DATA ACCESS MODE:
Table or View

NAME OF THE TABLE OR THE VIEW:
[dbo].[Student].

COLUMNS.

→ OK

Sort X

Sort Transformation Editor

STAGE

INPUT COLUMN - OUTPUT - SORT TYPE - SORT ORDER
STAGE STAGE Descending 1

→ OK

Destination Assistant.

→ Select Destination Type:

EXCEL

→ EXCEL Connection Manager.

EXCEL FilePath

OK

Browse

OLE DB SOURCE

↓↑ Sort

EXCEL DESTINATION

GoldenAddMS.

. EXCEL Destination Editor.

. EXCEL Connection Manager:

→ EXCEL Connection Manager.

. Data ACCESS Mode:

Table or View

. Name of the EXCEL Sheet:

grid mystudents.

. Create Table

. CREATE TABLE 'mystudents'

'ST_ID' Long,

'ST_FNAME' LongText,

'ST_LNAME' LongText,

'ST_Address' LongText,

'ST_Age' Long,

'DEPT_ID' Long,

'ST_SUPPER' Long}

MAPPINGS destination

→ INPUTCOLUMN - Destination COLUMN

ST_ID ST_ID

→ OK

. Not same version.

. integration services

project 13 property pages.

. Configuration Properties.

. Debugging

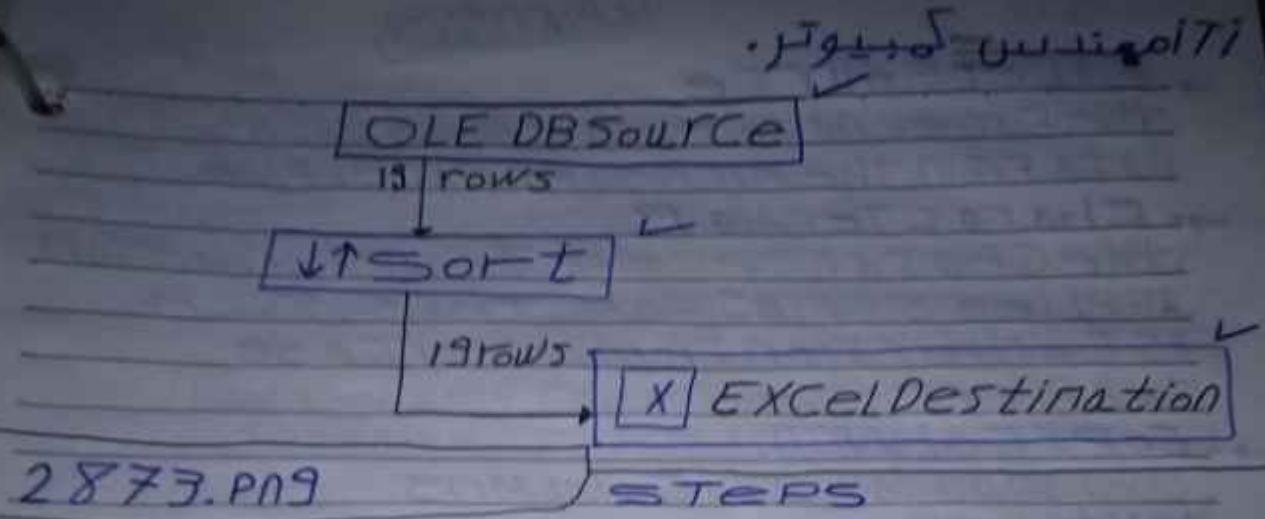
→ Debug Options:-

RUN64BITRUNTIME: FALSE.

→ OK.

→ . integration services project 13.

→ properties.



- Microsoft SQL Server
- New Database.
- Database Name: MYDW
- OK
- PACKAGE5.dtsx [Design]
- Favorites.
- DataFlow TASK.
- Source ASSISTANT - ADD NEW SOURCE.
- Select SOURCE TYPE:
- EXCEL.
- EXCEL CONNECTION MANAGER.
- EXCELFILERPATH:
F:\MYSTUDS.XLS
- OK
- TABLE OR VIEW.
- NAME OF THE EXCEL SHEET:
- mystudents.
- SELECT COLUMNS.
- OK

IN ENGLISH

- **EXCEL SOURCE**
the EXCEL Data Source obtains data from the Microsoft EXCEL file.
- **CHARACTERMAP**
the CharacterMap transformation applies string functions, such as conversion from lowercase to uppercase.
- **COPYCOLUMN**
copying input columns.
- **AUDIT**:
Audit transformation a PACKAGE Data About the ENVIRONMENT in which the PACKAGE runs. the NAME OF the PACKAGE, COMPUTER, OPERATOR.
- **DATA CONVERSION TRANSFORMATION**:
The Data Conversion converts the data in INPUT COLUMN to a different data type & then copies it to a NEW OUTPUT COLUMN.
- **DERIVED COLUMN TRANSFORMATION**:
Create new column values by applying expressions to transformation.
 $\text{FirstName} + '' + \text{LastName}$.
- **SORT TRANSFORMATION**:
The Sort sorts input data in ascending or descending order.

MS Learn.

- **READ BY YOURSELF**. Learn SQL
- SQL Server Integration Services.
- EX. CharacterMap, CopyColumn, Audit, DataConversion, DerivedColumn, Sort.

2912

900



- CharacterMap transformation
EDITOR:

Select ST Frame. V.
INPUTCOLUMN-Destination-operation-OUTPUT.
STFrame NewCOLUMN UPPERCASE COPY

- Derived COLUMN transformation Editor.

DERIVEDCOLUMNNAME-DerivedCOLUMN, EXPRESSION.
FulLName Laddas NewCOLUMN [ST_FName]+[
ST_LName]

→ [ST_FName]+[ST_LName]
OK

- COPYCOLUMN transformation EDITOR.

DePT_ID

INPUTCOLUMN-OUTPUTALIAS.

DePT_ID Did2

- Sort transformation EDITOR.

DEPT_ID

INPUTCOLUMN OUTPUTALIAS Sort type.

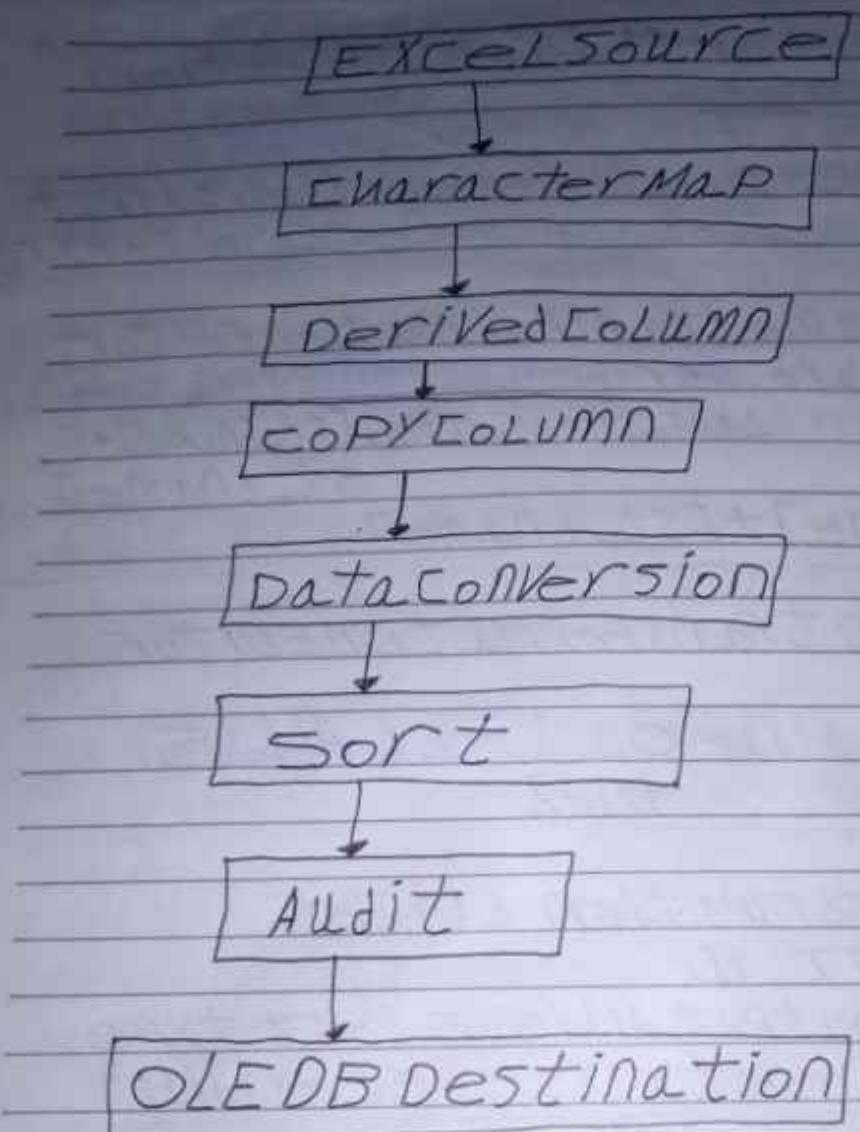
DePT_ID DePT_ID ascending

- Audit transformation EDITOR.

OUTPUTCOLUMNNAME Audit type.

Username Username v.

OK



- Destination Assistant -
ADD NEW DESTINATION.
- SELECT DESTINATION TYPE: SQL SERVER.
- CONNECTION MANAGER.
- SERVER NAME:
 • SELECT or ENTER A DATABASE NAME.
 MYDW
- OK

جسس III لف

- OLE DB Destination Editor.
- LocalHost.MYDW
- Table or view
- Name of the table or the view:
→ Create TABLE [FullStudentData] (New
.....
.....)

OK

PACKAGE.dtsx

→ DATAFLOW TASK

2 CLICK

- SOURCE ASSISTANT - ADD NEW SOURCE.
SQL Server.
- Connection Manager.
- Server Name
- Select or enter a database name
ITI
- OK
- OLE DB SOURCE EDITOR. ←
- OLE DB CONNECTION MANAGER:
- LOCALHOST.ITI New
- Data Access Mode:
Table or view
- Name of the table or the view.
[dbo].[Student]
- OK

Conditional SPLIT
Conditional SPLIT
Transformation Editor
order, output name, condition
1 LESS than REPLACENULL(STATE)
2 Greater than <23
REPLACENULL(STATE)
>23

OK

Condition:-
REPLACENULL([State], 0) < 23
REPLACENULL([State], 0) > 23
OK

INPUT OUTPUT SELECTION.

→ OUTPUT.
Less than

→ INPUT:
FlatFile Destination Input
OK

EXCEL Connection Manager

→ FlatFile Destination Editor.

→ FlatFile Format:-
Delimited

OK

Destination

EXCEL Destination Editor.

* EXCEL Connection Manager.

* Table or view.

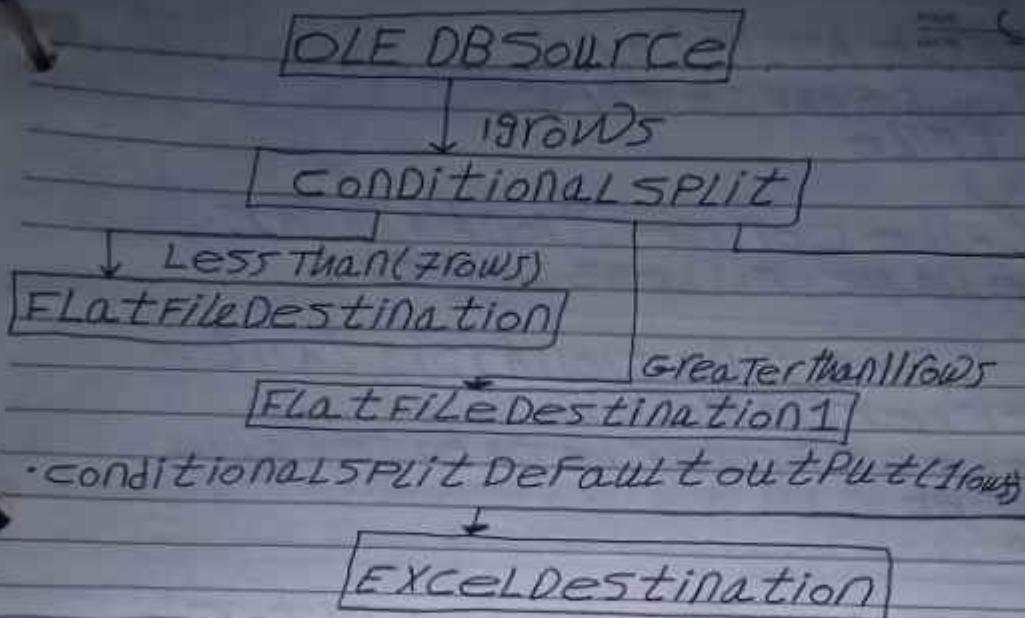
Name of EXCEL Sheet:

stud

OK

تطبيق على

العلى



Double CLICK . Enable Data viewer.

. Greater than Data viewer at Data FLOW TASK.

Less than Data viewer at DataFLOW TASK.

□ New SSIS Package.

Favorites.

Data FLOW TASK. Double click.

1. SOURCE ASSISTANT - ADD New source.

FLATFILE

FLATFILE Connection Manager File. EDITOR.

. FileName: Browse greaterthan.txt

Format: Delimited.

OK

FLatFile OK FlatFile source

source → destination.

2. SOURCE ASSISTANT

select source type:

FlatFile

New

FlatFile Connection Manager.

FileName: F:\Lessthan.txt

→ OK.

. Destination Assistant ::

ServerName:

● Select Enter Database Name:

MSdb

OK

OLE DB Destination. EDITOR.

• Create Table

• CREATE TABLE [Allstud5](

.....)

OLE DB Destination EDITOR.

→ LocalHost. MSdb

→ Table or View.

Name of Table or View:

[Allstud5]

FlatFileSource

FlatFileSource1

11rows

Union All

7rows

18rows

OLE DB Destination

2024/8/25 17:29

~~With Girlfriend.~~

Romantic Love

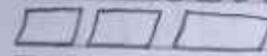
- BI DAY 2:
- Integration - Analysis - Reporting.
- ← Integration.
- Analysis.
- Reporting.

SSIS integration

ETL-ELT, ETL or ELT. أسلوب تحويل تحويل

ETL (Extract, transform, Load) tools
are required to ensure data is
integrated between external sources
and Microsoft SQL Server.

. Data Sources.



SQL Server, Oracle.

↳

↳

XML

Binary

SSAS.

Analysis.

FACT

Marketshare

Data Stage

DW.

DenormalizedDB,
HistoricalDB.

STAR Schema.

TIME	FACTS	PRODUCT
2000	Sales	
2001	Profit	
2002		CUSTOMER

BRANCH

EMPLOYEE

CUBES

With Girlfriend.

Romantic Love

BI DAY 2:

- Integration - Analysis - Reporting.
- Integration.
- Analysis.
- Reporting.

SSIS

Integration

ETL - ELT. ETL or ELT.

أين ترجمة تحويل تحويل

ETL (Extract, transform, Load) tools
are required to ensure data is

integrated between external sources
and Microsoft SQL Server.

Data Sources.



SQL Server, or Oracle.

XML

Binary

SSAS.

Analysis.

FACT2

Marketshare

Data Stage

DW.

Denormalized DB.
Historical DB.

STAR Schema.

TIME

2000

2001

2002

Facts

Product

Sales

Profit

Customer

Branch

Employee

Cubes

2024/8/25 17:22

اللمس

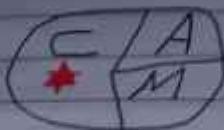
COFILES

البيانات

Year 0



Reporting.



SSRS.

Reports.

Dashboard.

*	*	*

step 1: integration.

step 2: ANALYSIS.

step 3: Reporting.

لوج

آندر ماج

تطبي



اعداد التقارير

ع

in integration, ANALYSIS, Reporting.

All objects saved AS XML.

All file transaction is XML.

أهم شيء
استخد

HTML displays Data and describes the structure of the a webPage.

whereas XML stores and

transfers Data بخزن XML . بينما يقوم XML
البيانات ونقلها .

* BI. شغال ثم شحالك ثم شحالك . الأهم

حتى لو بقيت وزير آخر في فلوس . التجارة .

قادم في الأمارات .

سیار قانونی

BI DAY 2.

BUSINESS INTELLIGENCE (BI):

the technical infrastructure that collects, stores, and analyzes company data.

Enable organizations to make better decisions.

MICROSOFT ANALYSIS SERVICES.

SSAS DEFINITION:-

SSAS Definition:

SQL SERVER ANALYSIS SERVICES

[SSAS] IS ONLINE ANALYTICAL PROCESSING (OLAP) SERVER & AN ANALYTICS ENGINE USED FOR DATA MINING.

Allows IT professionals to break up large data volumes of data into more easily analyzed parts. (Divide & Conquer).

SSAS CUBE DATA STRUCTURE IN SQL SERVER ANALYSIS SERVICES (SSAS).

Aggregate FUNCTION is Mathematical Computation.

SQL AGGREGATION FUNCTION:-

→ COUNT, SUM, AVG, MAX, MIN, AVG(Average)

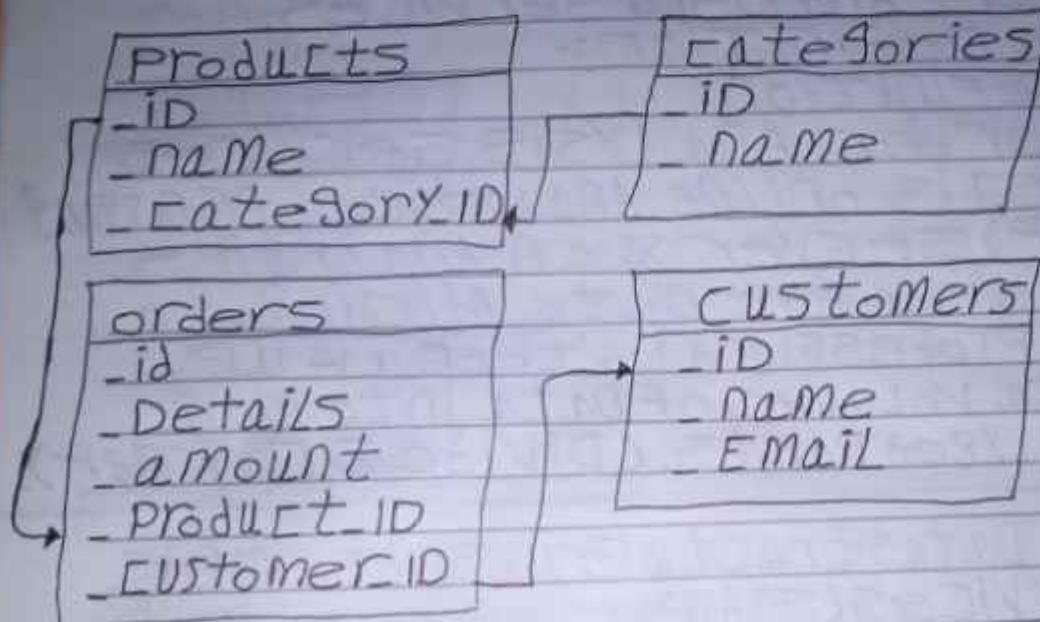
→ Microsoft Learn uses Aggregate FUNCTIONS.

CUBE: Calculations, Perspectives, Transformation. عمليات حسابية، تحويلات وpektives، Transformation.

Normalizing vs Denormalized Data.

Normalized :- القليل من البيانات المزدوجة

- Little to no redundant data.
- Many tables.



Denormalized :- افرقة عن الحالة

- redundant data is common.
- Fewer tables.

CustomerOrders	
.Id	
.Name_Product	
.Code	
.Name_Code	
.Email	
.order_id	
.order_Details	
.order_Amount	

2:28:55

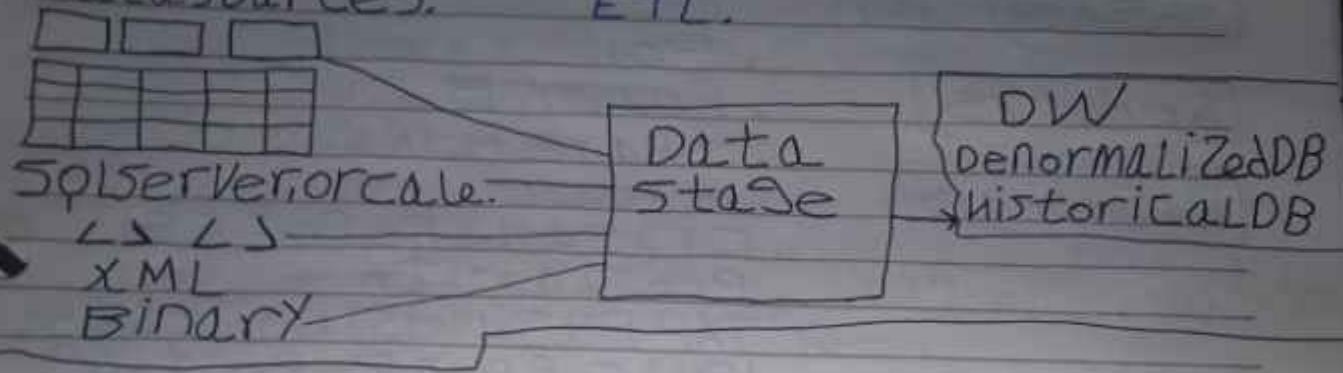
Ahmed
ADDLIMS

teradata (inFormatica).

-SISX (5515) oracle (DW builder).

PowerQuery.

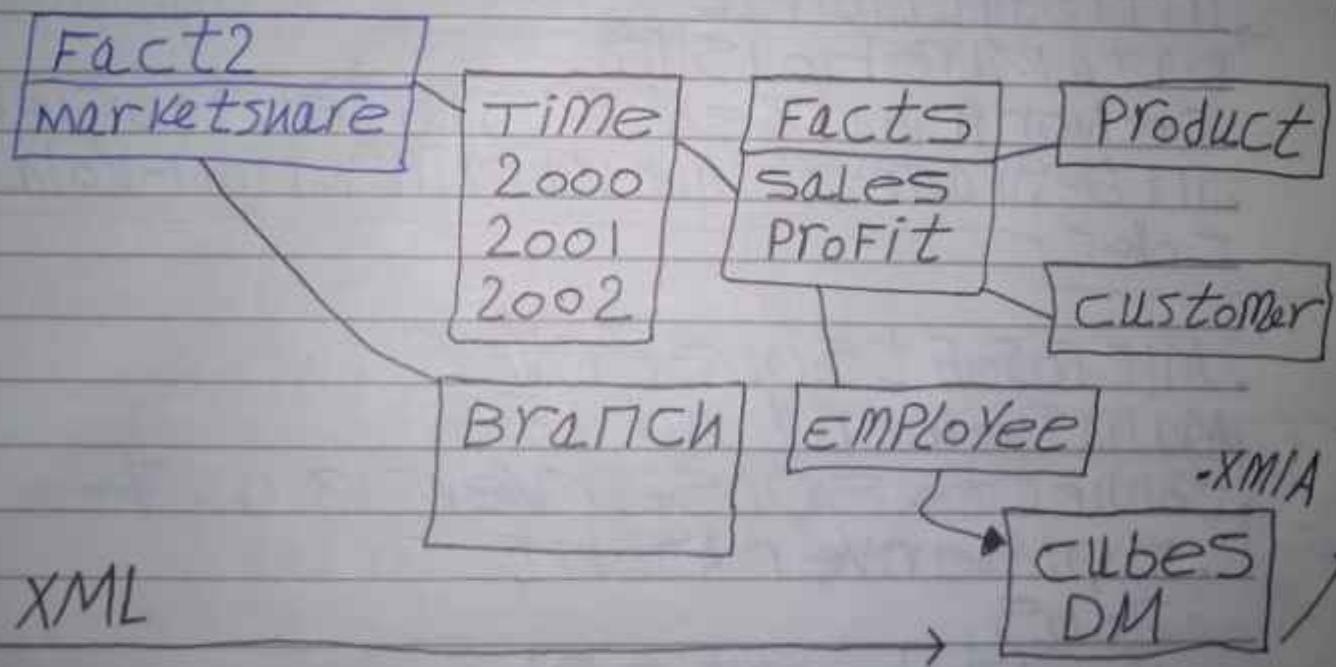
integration. ETL.
DataSources. ETL.



(55A5) PowerPivot

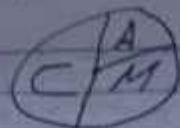
IBM Framework Manager.

(ANALYSIS) star schema.



خادم في الأمارات

Year ٢٠٢٣



Reporting

SSRS.rdl

Reports

PowerView.

PowerBI.✓

SharePoint.

Excel.✓

CrystalReports.

ProClarity.

✓ SAP BusinessObjects.

IBM Cognos Reporting.

Tableau...✓

target BI.✓

→ introduction to

Data Warehousing.

DeNormalized DB Subject-oriented

Integrated, time varying, non-Volatile
Collections

Microsoft SQL Server

Management Studio.

Connect → SQL Server 13.0.17

SQL Server Agent

Jobs

New Job → T-SQL

New Job Step

1. StepName: S1

2. Type: SQL Server Integration Services

3. RunAs: Package

4. OK

2024/8/25 17:31

• New Job → Schedules → New Job Schedule.

• Data warehouse v/s OLTP.

→ run services [local].

• SQL Server (MSSQL Server). service

→ SQL Server (MSSQL Server).

• to summarize... لذلك

• OLTP systems are used to run a business.

• online transaction processing:

→ online banking - shopping - order entry

- sending text messages.

• The Datawarehouse helps to optimize the business. يساعد في تحسين الأداء.

• Two Dimensional Cube:-

	C1	C2	C3
P1	12		50
P2	11	8	

• 3 Dimensional Cube:-

Day2	P1	C1	C2	C3
Day1	P1	12		50
	P2	11	8	

ترافق rollup.
أنتقل drill-down. إلى أسفل

Example: computing SUMS:-

	C1	C2	C3
P1	56	4	50
P2	11	8	

	C1	C2	C3
SUM	67	12	50

→ 129

	SUM
P1	110
P2	19

MICROSOFT SQL SERVER M STUDIO

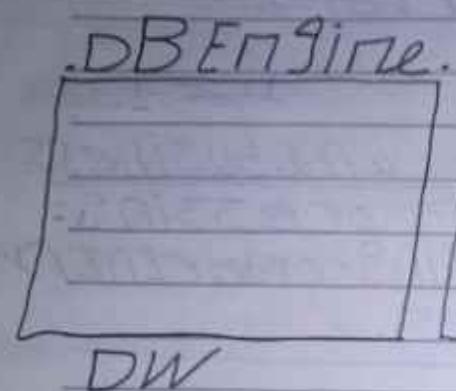
. SQL Server 13.0.1742

. Databases

. MYDW

- Tables

dbo.FL115tudData



OLAP

ANALYSIS SERVICE

Reports

High

Cubes

[DIM Data]

+ Aggregates

↑ Performance

HOLAP

.ROLAP ← Cube → structure
Low Performance

Performance.

.SQL Server:

→ ANALYSIS SERVICES (MSSQL Server).
SERVICES (LOCAL).

. MULTIDIMENSIONAL PROJECT 13.

. FILE → NEW ► Project

. New Project ► Installed

→ BUSINESS INTELLIGENCE

. ANALYSIS SERVICES

. ANALYSIS SERVICES MULTIDIMENSIONAL

* Data Source Wizard.

. Data Connections.

→ LOCAL HOST ITI.

(New)

العنوان

Connection Manager.

. Servername:

All authentication: Windows Authentication.

. OK

. Select or Enter Database Name

. Next

. Data Source Wizard

. Use the Service Account

. Data Source Designer

Data Source Designer

→ Use a Specific Windows Username & Password.

. Use the Service Account

OK

. Data Source Views.

. New Data Source View.

i. Data Source View Wizard → Next

Select Data Sources: Next

Select Table and Views

Name: FactSales(dbo). Next

. Data Source View Wizard:

→ Name: Sales. SALES

. FINISH.

. FactSales

. ProductDim

. TimeDim

Sales.csv [Design]

Diagram or Organizar

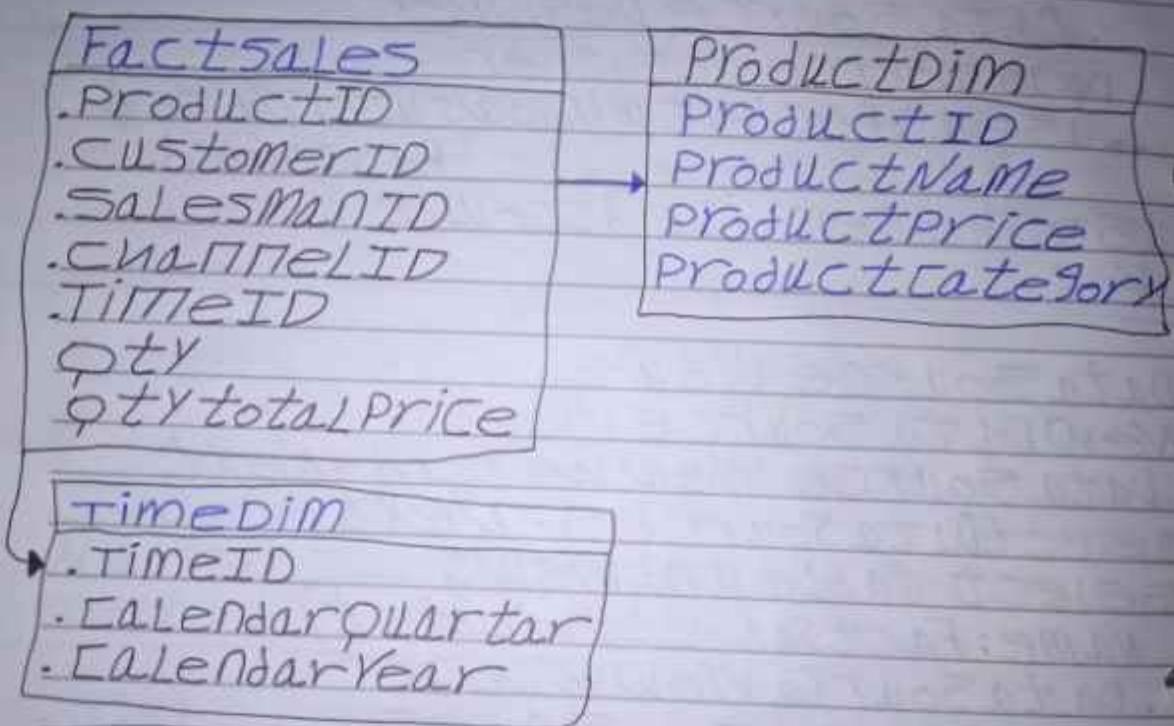
(All Tables)

Tables:-

+ FactSales

+ ProductDim

+ TimeDim



DataSources:

Cubes → NewCube...

CubeWizard

→ Use EXISTING tables. **Next.**

Data Source View: SALES

Measure group tables:

✓ FactSales. **Next.**

→ CubeWizard → Select Measures.

✓ FactSales.

✓ Qty.

✓ QtyTotalPrice.

✓ FactSalesCount.

Next

خطف آخر ٤٨ منس

ENGLISH

Select Measures.

✓ FactSales

✓ Qty

QtyTotalPrice

Next

FactSalesCount

Select New Dimensions.

→ ✓ Dimension.

✓ ProductDim. Next.

→ Cubewizard → Cubename: SALES. FINISH.

.54) Server Analysis Services.

studyonline ✓

Aggregate Functions

.55) :- Data Mining Group of Algorithms

Applied on Data.

. Aggregate Functions in SQL.

. Algorithm is used in Data Analysis?

regression - Classification - Clustering

- Decision trees - random forests &

Machine Learning techniques

Like Supervised - Unsupervised -

Reinforcement Learning.

BI

. Business Intelligence is SSAS Cubes.

step by step.

شرح في بى بى واحد.

تسلسل ورقة واحدة بودرة.

تبين الخطوات في الآخر.

. ITI Business Intelligence BI.

شرح في

Business intelligence BI
SSIS SSAS cubes SSRS.
BIDAY3.
Project, ٤

تحليل البيانات العملية وذكاء الأعمال.

Data ANALYSIS.

Analyzed to answer questions.
Decision-making.

BUSINESS INTELLIGENCE.

BIG DATA: A REVOLUTION THAT WILL
TRANSFORM HOW WE LIVE,
WORK, AND THINK.

FOURTH INDUSTRIAL REVOLUTION.

الثورة الصناعية الرابعة

What is 'Business Intelligence'?

ما هو ذكاء الأعمال.

جمع كمية كبيرة من البيانات لاتخاذ القرار.

و تصور البيانات.

مقارنة البيانات هذه السنة و السنة الماضي.

أداء الأطوال النسبية.

Collect a large amount of data for
decision making and data visualization.

Ex:- Comparing Sales this year
and last year.

* تحسين التائج والمبادرات.

MAKE DECISIONS FASTER.

BUSINESS INTELLIGENCE.

1. الاتصالات التي تحلل وتحوّلها بجهزه في كل وقت.
2. دمج مصادر البيانات متعددة في تقرير واحد.
3. تصفح التقارير عبر الانترنت على جميع انواع الأجهزة.
4. تحديد بياناتك بشكل آمن، ومتابعة مؤشرات الأداء.

MICROSOFT SQL Server 2019 Reporting Services. Downloaded.

- BI Advantages over standard reporting.
 - . INSIGHT:- Making the right Decisions.
 - . ACCESSIBILITY:- View Reports online.
 - . POWERFUL ANALYTICS:- Answers you want.
 - . CONNECTIVITY:- Direct Link to Database.
 - . DATA INTEGRATION:- Combine multiple Data Sources in one Dashboard. (SQL, SAP, EXCEL, ACCESS...). SSIS.
 - . VISUALISATION:- [Charts and tables.]

تحليل البيانات العمليه وذكاء الاعمال. KPIs
تحليل البيانات وذكاء الاعمال. (BIA).
MBA Entrepreneurship. دورات
تحليل البيانات وذكاء الاعمال عبارة عن مصطلح
يترجمه البعض من قبل واكتب هنا تعني عملياً في
ذكاء الاعمال.
في هذه الدورة التدريبية المصغرة، سنطرق
الي مفهوم يمكن من خلاله تحقيق تحليل البيانات
 بطريقة عملية وأصلع من خلال العمل.

ما هو علم البيانات (Data Science).

أو في هذا المعلوم البيانات: أنت تخدم
البيانات لحل المشكلات العمل أو غير ذلك.

- USE Data to SOLVE BUSINESS problems?
- * DATA SCIENCE Extract KNOWLEDGE
and INSIGHTS FROM your DATA.

Data Science.

STATISTICS & COMPUTER
MATHEMATICS | PROGRAMMING | DOMAIN
AS Netflix SEARCH & VIEW FILMS

ما هو تحليل البيانات / نكاء الأعمال (Data Analysis/BI)

المهمة الأساسية للأجابة على الأسئلة

- عملاء باستخراج البيانات
- To ANSWER YOUR BUSINESS
QUESTION.
- Sales.

هل هنا أعلم ببيانات؟

يعتمد على من تأسئلة.

يستخدم البيانات للإجابة على الأسئلة.

BI (Business Intelligence) نكاء الأعمال.

إنه لا يستخدم النتائج التنبؤية

أو التعلم الآلي Predictive Modeling
أو الذكاء الاصطناعي AI
Machine Learning
(أكثر وصفاً في التوجيه).

Data Science → NETFLIX.

BITCOIN CRYPTO Chart → BI Data Analysis.

2015 2016 2017

2024/8/25 17:45

تحليل البيانات

- Data ANALYSIS. تحليل البيانات
- * STATISTICS & Mathematics. لاحصاءات و الرياضيات
- * Computer Programming. البرمجة
- * Domain Knowledge. برمجة الحاسوب

تعريف المجال

يمكن أن نسمي "تحليل البيانات" "علم البيانات" فهو من نوع
أو بترجمة "علم البيانات" أو "الآدوات الجاهزة"

من أصل:

- 1) العمل بسرعة ✓
- 2) العمل بفعالية ✓
- 3) عدم تخفيض الموارد. ✓

- Data ANALYSIS No Code Data SCIENCE.
- على هذا ينبع سؤال .
- Which one is easier to start with?
- عليها أن تكون أسلوب انتبه لها
- Create without code: the No Code Landscape.
- Create without code.com
- EX: Data ANALYSIS tool PowerBI.

Big Data * like Facebook Likes.

* Google search results.

الناس يحبونه و يتقنه.

تحقيق هناعي جانب البرمجة لعلوم البيانات (مستويات التجربة).

علم البيانات Data Science

1. statistics & Mathematics.

أحصائيات الرياضيات.

Machine Learning التعلم الآلي

2. Computer Programming.

برمجة الحاسوب

البيانات الخروجية Automation / Big Data

3. Domain Knowledge معرفة المجال

Business Expertise الخبرة التجارية

لعونا ونوضح كيفية أسلوب اهتمام البيانات عملية لفهم المراحل الخمس.

Goal → Collection → Processing

→ Analysis → Decision.

الخطوة 1: المعرفة Goal.

سؤال: ما هو العائد على المستثمار API الكامل.

لوجودنا التسويقية.

حل البيانات: ماهي القيمة الدائمة للعمل.

(الربحية الفعلية) لكتبة تجارة تسويق.

الخطوة 2: الجمجمة Collection

بيانات الخاصة ✓.

بيانات الشركاء ✓.

بيانات العملاء (Online/Offline).

بيانات الطرق الثالث.

بيانات على الإنترنت / بيانات التتبع.

بيانات الخروجية (API) والتكاملات أخرى *

توبه

Collection

بيانات الشركة :-

معلومات الشركة ✓

الشراء والمخزون ✓

المعلومات اللوجستية ✓

بيانات التدفقات ✓

على سبيل مثال، تريد جمع بيانات التدفقات.

بيانات العملاء.

بيانات الشراء.

بيانات العميل.

بيانات الإيرادات.

برامج الواتس collection

على سبيل مثال، تريد جمع بيانات العملاء والإيرادات.

بيانات العملاء عبر الانترنت :-

1. بيانات التجارة الإلكترونية.

2. بيانات قنوات التسويق.

3. الأسلواع في الموقعا.

على سبيل مثال، تريد جمع بيانات قنوات التجارة الإلكترونية والتسويق.

* كيف نجمع هذه البيانات؟

Datatable

هي مكتبة تكون:

تم تجميعها أصلًا "دويا" (Excel)، جداول بيانات.

قواعد البيانات الائتمانية (أنتقامية) لإدارة علاقات

CRM.

* قواعد بيانات العملاء.

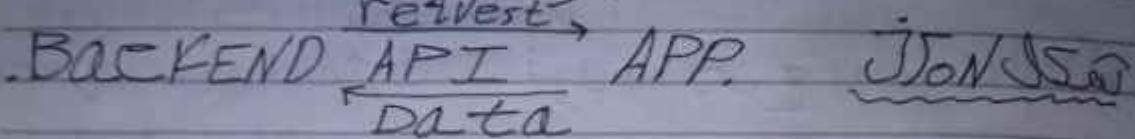
* برامج BI الداخلي.

* البيانات الخارجية (في السحابة - Cloud).

'EXCEL BASIC Data Analysis.'

GentelMan

أختبار البيانات الخارجية
واجهة برمجة التطبيقات: واجهة
برمجة التطبيقات.



API GoogleAnalytics: API لـ *
 Import requests. ✓
 Requests.get(). ✓

GoogleAnalytics:
 ANALYTICS TOOL & SOLUTIONS
 For Your BUSINESS.
 GoogleAnalytics API.

ما هي البيانات التي احتاجها
 - (قيمة عمر العميل)
 - بيانات التكاليف
 - بيانات العميل.
 - بيانات التسويق.

Year	Month	Channel	cost
2018	October	Facebook Ads	10000

Email	Revenue	order	customer
ad@gmail.com	121	1	1

Year	Month	Source	Medium	transaction	ID
2018	October	Facebook	Ad	1	2024/8/25 17:46

كيف نعالج هذه البيانات؟
برنامج BI: يأخذ كل جم ومنظمه ومتغيره

أبرواترون آتصال:
SAS-oracle BISuite - AZURE.

عبر الانترنت (على Cloud).
OFFICE 365 Power BI For OFFICE 365.
Power BI - Google Data studio - QLIK -
Tableau

مصادر رائجة:
Google Analytics, BI Power,
Google Ads, Doubleclick.

الخطوة 3: المعالجة
المعالجة.

كل نوعي.
عمل الجدول الموردي.
عمل الـ Pivot table.
عمل التدوير العمر - الجنس.

insert chart

Pivot table Excel في Chart

EX:-

Data Analysis improve Daylife

PK → FK :-
المفتاح الأسياسي (يحد الجدول الأصل)
المفتاح الأسياسي (يحد الجدول الأصل)
وهو مشترك في كل جدول الفرعية.
المفتاح الثانوي (يحد جدول الفرعية).

* أمثلة من :-
بيانات غير صالح (ملف خارج، مفقاعة).

مكرر

رقم الجداول.

Master table

الجدول الرئيسي

CustomerData
بيانات العميل

CostData
بيانات التكلفة

MarketingData
بيانات التسويق

Year	Month	Channel	Cost
2018	October	FacebookA	10000

Year	Month	Source	Medium	transactionID
2018	October	Facebook	Ad	1

Email	Revenue	order #	customer #
l@9mail.com	121	1	1

* ربط المنهجيات مع البيانات.

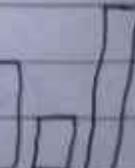
الخطوة: التحليل
use tables & charts:-



Pie.



Bar



Column

برمجة البرمجة

• Some types of chart:-

Pie, Bar, Column, Line, Area,
Doughnut, Bubble chart, Spider and Radar
, scatter, comparison chart,
Stacked bar chart, Gauges

على حسب ترتيب الدراسات
. charts . ترتيب

* الخطوة 5: القرار Decision . هذان يحول إلى خط
verbal part . Decision . هذان يحول إلى خط

Goal - Collection →
processing → Analysis → Decision.

. 1 Goal → 2 Collection → 3 Processing
→ 4 Analysis → 5 Decision.

* Domain Knowledge . معرفة المجال

. Next step :- What is the next step?

- . Data Scientist
- . Data Analyst
- . Business Analyst

Data Analyst &
Power BI Developer

- Read about this Fields:-
- Western Union Send Money Worldwide.
- Entrepreneur Bill Gates.
- Data Science.
 - SAM: Made in America / Same One Book
 - Same one book.
 - Project. إبراهيم عباس.

- BT - Data warehousing - AT Data + Science - Microsoft Server
- EntrepreneurSHIP . إبراهيم عباس.
- Startups Stories.

- BUSINESS ANNUAL San Jose Salary . ممثل زنار في جي بي

- Read About Deepfake & New technology.

- Aries . Richest . Larry Page .
- Twinkie Brothers . Jerry Brin . Venus Opposition Pluto .
- Mars in Capricorn . Venus Conjunction Pluto .
- Jupiter fine Pluto .

- End Make Sheet Documentation . Work with Jupiter to book Sator .
- No Study with Jupiter Project Sator . Applied .
- it's revision make Project Sator .
- writing . Screen shot video . Project

OLAP: online analytical processing
OLAP: Software technology you can use to analyze business data from different point of view
Collect & store data from multiple data sources.

Websites, Applications, Smart Meters, Internal Systems.

DeNormalization is used in OLAP system on making the search and analysis faster.

Make cube than analysis than report.

one table في tables و join.
→ Normalization to DeNormalization.
11 table into 6 tables.

اجماع

DIMCustomer بحسبية Fact البيع DimDate.
DIMProduct بحسبية DimTime.

بالنسبة لـ fact العميل الامريكي.

→ FPTIS → target الفعل. → true.
الغرض نسبه مبيعات 2000 → false.

Integration → Packages → DataWarehouse.

Fractional Data Analysis & ETL

- Online transaction Processing OLTP.
- Normalization is used in OLTP system, on making the insert, delete and update anomalies faster.
- High performance & speed.
- Data warehouse is used in OLDW.
- Used in Build Website.
- → Online banking, shopping or sending text messages.

هنا أتعلم

.tabular Report
.Matrix Report
.chart Report
.FreeForm Report

* EXAM.

* COURSE + TOPICS.
* EMAIL.

1	2	3	4	5	6	7	8	9	10

Pivot



chart

Deployment.

H[HTTP://localhost/Reports])

.SQL Server Data Tools →
Design Reports (.rdl) →
Physical Files,

Deploy

SSRS [HTTP://localhost/ReportServer]

SSMS [Report Server]
[ReportServer tempDB]

Website:-

HTTP://localhost/Reports

.SQL Server 2016:

→ Report Services Configuration
Manager:-

.Web Service URL.

MUST Reporting services RUN.

USE StudentReport;
CREATE PROC GetStuds
AS
SELECT *
FROM Student
WHERE Student = 'Cairo';
DATASETS → DATASETS1 → ADD CALCULATED FIELD
→ QUERY → QUERY DESIGNER... → COMMAND TYPE:
→ STORED PROCEDURE →

DATASETS1 → RIGHT CLICK →

* CREATE PROC GetStudsByAge1
@Age1 int, @Age2 int

AS
SELECT *, (SELECT AVG(Age) FROM Student)
AS AvgAge
FROM dbo.Student
WHERE Student.Age BETWEEN @Age1
AND @Age2;

EXEC GetStudsByAge1 20, 30

1# CREATE PROC GetStuds

AS

SELECT *

FROM dbo.Student

WHERE Student.Address = 'Cairo';

EXEC GetStuds

USE StudentReport;

SELECT * FROM dbo.Student;

on SQL Server

• Data Source → Datasets.
Datasets → Dataset1 → right click
→ ADD CALCULATED FIELDS →
Dataset Properties → Query
→ Query Type: → STORED PROCEDURE →
Select or enter stored procedure name:
GetStudentbyAge1
→ Query Designer → !
* Define query parameters
@age1 ✓
@age2 ✓
OK

FIELDS → OK. Parameters
 Datasets1 age1
DATASET1 age2

ID
Name
Address
Age
DepartmentID
AVGAge

DESIGN

age1

age2

VIEW REPORT

STUDENT REPORT

ID Name Address Age AVGAge DepartmentID
[ID] [Name] [Address] [Age] [AVGAge] [DepartmentID]

• Report with parameter.

→ Properties Pages.

• Start → Create Linked Report.

• Microsoft SQL Server 2019 Reporting Services → YouTube.

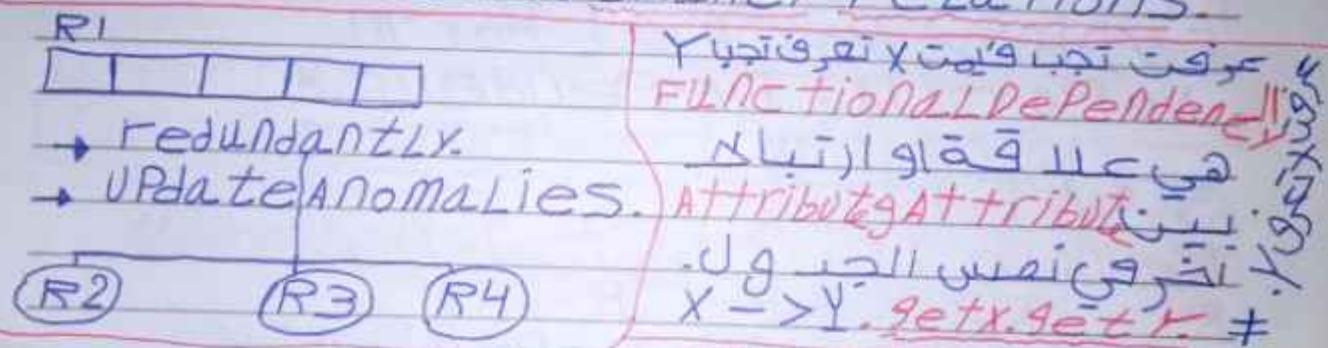
الآلات والمعدات وأجهزة الكمبيوتر.

Database Normalization.

- CHAPTER 10. DATABASE SYSTEMS.
 - FUNCTIONAL DEPENDENCIES AND NORMALIZATION FOR RELATIONAL DATABASES.
Error in:- No this in Data.
Redundantly.
Update Anomalies
(Insert-Update-Delete).

- Normalization. No redundancy ✓
 - No Update Anomalies ✓

Normalization:- breaking up their attributes into smaller relations



EXAMPLES OF FD CONSTRAINTS:

Social Security Determines Employee Name

~~SSN->ENAME. SSN is a Global.~~

نامه فرماندهی نیروی انتظامی

ESSN, PNUMBER, & -> HOURS.

- Normal Forms Based on Primary Key
- First Normal Form. Second. Third.
Boyco Codd. Fourth. FIFTH. Normal Form.

1st NF

Remove MV Attributes.

R1. multivalue.

→ ISSUES:-

Repeating Groups Redundancy,

Partial, transitive Dependencies

→ ENDING 1st NORMAL FORM :-

R1 R2

• No Repeating Groups.

• No MV Attributes.

2nd NF

Remove Partial Dependency.

R1 R2 R3

→ ENDING 2nd NORMAL Form.

• No Partial Dependency

3rd NF

Remove transitive Dependency.

R1 R2 R3 R4

→ No Transitive Dependency

→ DN Normalization.

→ First Normal Form.

جني يجب أن يصبح الجدول في الـ 1st NF.

→ composite Attribute, MV

Attributes and Nested Relations.

① Department

DName | DNumber | Manager | DLocations

zero
form



Report1.rdl[Design]
Design

العنوان

Jobs TITLE SALARYS

ID	JobTitle	MinSalary	MaxSalary
[JobID]	[Job_title]	[min_salary]	[max_salary]

[&ExecutionTime]

[&Language]

By SE:- Jacob's

→ Preview

Export ▾

Word

Excel

PowerPoint

PDF

TIFF File

HTML.

MHTML (Webarchive)

CSV

XML File With Report Data.

FirstReport.mhtml

VISUAL Studio 2022 (SSIS, SSAS, SSRS).

SQL Server.

STUDYNow 'BI SSRS'.

Screenshots.

يحتاج تذكرها لازم نعملها

Documentation.

لَا عَانِي تُمْ لَا
كُدُّتْ قَلَّةَ مَنْ النَّاسُ
تَحْوِلْ حَاجَةَ اَسْعِدْنَا

* SSRS WITH Parameter.

- MICROSOFT SQL SERVER MANAGEMENT STUDIO → CONNECT → OBJECT EXPLORER.
- MAKE THE SAME REPORTS.
- SAME DESIGN.

→ NEW PROJECT → REPORTING SERVICES
→ REPORT SERVER PROJECT WIZARD.
→ OK.

1. SELECT THE 'DATA SOURCE'.

- CONNECT TO A DATABASE.

2. DESIGN THE QUERY.

- USE QUERY BUILDER...
- QUERYSSTRING.

3. SELECT THE REPORT TYPE.

- TABULAR. DESIGN THE TABLE.
- MATRIX

□ Mansoura Reports. → right click

→ PROPERTY PAGES → start item.

General → DEBUG: → start item
choose: Report1.rdl → you want run.

• EXPRESSION. . VIEW.

• right click → ADD PAGE HEADER.

• ADD PAGE FOOTER.

image properties.)

• CLICK ON ROW THEN PROPERTIES.

→ BACKGROUND COLOR: Light Green.

الأخوان دايت.

Report1.rdl[Design]

Design | Preview

IT Mansoura

image

STUDID	Name	STAddress	STAge	DEPTID
[ST_ID]	[LEXPR1]	[STAddress]	[STAge]	[DEPTID]

[&UserID] [&ExecutionTime] [&PageNumber]

SID Name Address Age DId Dollar

.create Database.
.SQL create Database.
.CreateDatabase STUDENTREPORT;
.EXECUTE
.REFRESH
.USE STUDENTREPORT;
CREATE TABLE STUDENT(
ID int primarykey NOTNULL,
Name char(50) NOTNULL,
Address Char(50) NOTNULL,
Age int NOTNULL,
DepartmentID int NOTNULL);
→ EXECUTE.

```
USE StudentReport;  
INSERT INTO Student  
VALUES(1,'A','Cairo',23,10),  
(2,'B','Mansoura',25,20),  
(3,'C','Cairo',28,30),  
(4,'D','Alex',23,40),  
(5,'E','LUL',26,50),  
(6,'F','Cairo',23,60),  
(7,'G','Cairo',32,70),  
(8,'H','EL MahaLaN',35,80);
```

Execute

• Refresh Database.

```
USE StudentReport;  
Select * FROM Student;
```

→ Result.

• MS VISUAL Studio 2022. → Create New Project.
→ Report Server Project Wizard. → Next
→ Project Name: studentProject → Create.

* Welcome to the Report Wizard. Next.

* Select the Data Source.

• New Data Source

Name:

Data Source student

• Type:

MICROSOFT SQL SERVER

• Connection String: • ServerName: •

Database Name:

StudentReport

testConnection

OK

Next

NoSQL

→ Design the query.

queryBuilder...

queryDesigner:

select *

from Student!

OK

queryString

select * from student

Next ↴

Select the Report type.

• Tabular

Next ↴

Design the table.

DisplayedFields <= Fields in .xmi.

ID

Name

Address

Age

DepartmentID

*

Next ↴

ReportName: StudentReport.

Finish

view → ToolBox.

rightCLICK → body → TextBoxProperties

Fill → FillColor: BLACK. OK.

ID → rightCLICK → TextBoxProperties.

Font. Color: Red. or Blue.

. Effect: Underline.

OK

Image

الجامعة

Parameters.

Δ Images. rightCLICK → ADD IMAGE...

chooseImage. JPEG

. View → toolbox → Image.

. View → Toolbox → textBox → rightCLICK
→ Create Placeholder... → Value: → FX →
Built-in Fields → UserID → = User!UserID.
Value: [&UserID] → OK

. Built-in Fields → ExecutionTime → Double
Expression. CLICK
= Globals!ExecutionTime. → OK.
OK

. View → toolbox → textBox.

. Placeholder → Value: → FX → Built-in Fields
→ PageNumber → OK.

Export MHTML (WebArchive). Header

iTIMANSOURA STUDENTREPORT

ID NAME ADDRESS AGE DEPARTMENTID
[ID][NAME][ADDRESS][AGE][DEPARTMENTID]

the New
America
College

Footer

[&UserID][&ExecutionTime][&PageNumber]

STUDENTREPORT.rdl [Design]

الـ 1.1.1 **Collaboration** . مـ 1.1.1

ـ مـ 1.1.2 **Partnership** .

ـ مـ 1.1.3 **Cooperation** . جـ 1.1.3

ـ جـ 1.1.4 **Merger** . حـ 1.1.4

ـ حـ 1.1.5 **Acquisition** . حـ 1.1.5

ـ حـ 1.1.6 **Joint Venture** . حـ 1.1.6

ـ حـ 1.1.7 **Family Business** . حـ 1.1.7

ـ حـ 1.1.8 **Succession** . حـ 1.1.8

ـ حـ 1.1.9 **Intergenerational Succession** . حـ 1.1.9

ـ حـ 1.1.10 **Jupiter-Cancer Taurus** . حـ 1.1.10

ـ حـ 1.1.11 **Jupiter-Orion** . حـ 1.1.11

BLACK ERIDAY

WATER. EGG. SALT.
FISH. SALT. SALT.

Job Insights

Based on previously hired candidates

Popular Skills Of Previous Hires

Business Intelligence

Business Analysis

Big Data

AI - Deep and Machine Learning
and Ar

Data Warehousing

Business Requirements

Job Titles Of Previous Hires

Data Engineer

Business Intelligence Analyst

Data Analyst

Senior Business Intelligence Analyst

BI Analyst

Previous Hires Also Worked At



[Apply Now](#)