

* Chapter three :-

1

Relational Model used instead of 1- Hierarchical Model
2- Network Model.

Idea :- Organize data as a group of Relations, each relation describes a group of object with similar attributes

* Schema : Description with-out data (few columns)

example students (sid : string, name : string, login : string
age : integer, gpa : real)

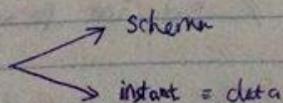
⇒ The previous schema says that each record in the students relation has five fields [field name & type]

* instance ≡ data
- redundant :  schema ⇒ sid | name | login | age | gpa
instance ⇒ 53666 | Jane | Jane@cs | 18 | 3.4

* We can Note that : Database now is a collection of one or more relations
each relation is Table

* We will use "MySQL" to 1- query 2- manipulate data 3- relations

* DML : Data Manipulation Language
DDL : Data definition Language

* Relation 

if Variable is PK, it define Not Null

3

- show Tables; // To show all tables in data base

- show create table students; // To show student's table without data

Ex How to show data ?

Ans Select * From Students;

Ex How to add another att. ^{columns} for table students ?

الحول بين < عین ایلایة ای الکر الایتم
duplicate

Two Solutions 1) drop table then rebuild

2) ^{تبدیل} ALTER Table Students add major Varchar(16);

Ex How to add the att (column) at certain place ?

ALTER Table students add major Varchar(16) after birthdate;

* Inserting data *

Insert Into Students Values (1152233, 'Ahmad', 1980-02-20, ...), ^{عین ای ای}

or Insert Into Students (sid, name) Values (... , ...);

Note int = Integer

Varchar (11) ⇒ 11 to 11

char (11) ⇒ 11

* Slides Example :-

- Delete from Students where name = 'Smith'

- UPDATE Students Set

age = age + 1, gpa = gpa - 1

where sid = 53688

این چهار تغییر بزرگ: %
 عیب مطلوب نیست
 PRIVILEGES :
 آشنای سلاطین ← دراست

Alert If this Message was appear

" You are using safe update mode and you tried to update a table without without ... "

Solution
 Edit → Preferences
 SQL Editor → uncheck " safe update "
 ok
 Reconnect to server

* drop table Students ;

* delete from Student where sid > 100 AND sid < 200

Integrity Constraints Over Relations

تقييد سلامة العلاقات

* IC :- (Integrity Constraints)

Is a condition that is specified on a database schema and restricts the data that can be stored in an instance of the database.

Key Constraints :- Statement that a specific minimal subset of the fields of a relation is a unique identifier for a tuple

المتطلب :- (1) أن كل Tuple (tuple) في العلاقة لا يكون له نفس القيم في مجموعة من حقولها (Key) (Label) (Constraints) (Constraints) (Label) (Constraints)

No subset of fields in Key is a unique identifier for a tuple (2) Key is minimal

* Keys :-

(1) Composite Key : Composed of more than one attribute
لا يكون اسم واحد من حقولها هو المتطلب بل مجموعة من الحقول
مثال : العناوين بحسب الأرقام و رقم الهاتف

(2) Key attribute : Any attribute that is a part of Key

(3) Super Key : Any Key that uniquely identifies each row
من خلال تغير قيم الحقول في العلاقة
بشكل فريد و اسم واحد

4) Candidate Key :- a super Key with out redundancies
نفس المبرر كفي من دون تكرار

Note <- Key من شئ يكون الكلي تابعي
الحجم اقل من اجزائها
unique access <=>

- NULL =>
- 1) No data entry
 - 2) Not permitted in primary Key عنى صفر يكون
 - 3) should be avoided in some attributes NULL <- PK
 - 4) can be represent @ an unknown attribute value
 - ⓐ a known "missing" attr. value
 - ⓑ not applicable condition

Note 1) Key is Minimal something
2) Super Key is Not Minimal (Key + sth)

Ex does Key is super Key?
=> T. Key always minimal

Ex does super Key is Key?
F, super Key is not minimal

* Sid integer not NULL ; // يجب استخدام Integer
يا براكه فاجبه

Example alter table students add major varchar (20)
not NULL after birthdate;

Remember:- Create table students (
 sid int Primary Key, ✓
 [of] sid int,
 Primary Key (sid), ✓
 sname varchar (20) not null,
 gpa real),
 89.9 جانب ←

Note Constraint EmpPK Primary Key (EmpID) // Prim. Key
 with const.
 it → Label

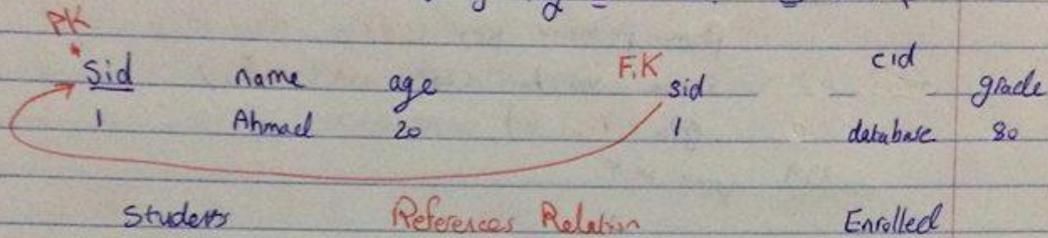
Note Constraint EmpPK Primary Key (EmpID, skillID)
 Composite Primary Key using constraint

Note a combination of PKeys \Rightarrow super-Key
 \Rightarrow a pair of entity can have at most one relation

PK: جانب Constraints ايجز لا تستعمل

Foreign Key :

* Key لجدول و يوجد به جدول ثانى
يسمى هذه Key فى الجدول الثانى
ف اسمها فى الجدول الثانى Foreign Key



Ex Write _____ enrolled table in MySQL ?

Ans create table enrolled (

sid char (20),
cid char (20),
Primary Key (sid, cid),

* Foreign Key (sid) References Students (sid),

FK (sid) ref students (sid) ✓ يمكن يربطها
FK (cid) ref course (cid) ✓ يمكن يربطها

IC (Int. Cons.) : state that every value of FK must match a value of an existing PK

Ex Example from slides on FK

Create table Emp-Skill (

EmpID integer not null,
SkillID integer not null,
SkillLevel integer,

Constraint EmpSkillID Primary Key (EmpID, SkillID),

Constraint EmpFK Foreign Key EmpID
References Employee (EmpID), ON Delete Cascade

Constraint SkillFK Foreign Key SkillID
References Skill (SkillID) ON update Cascade

→ if EmpID row deleted → PK
EmpFK is ~~deleted~~ deleted → FK

* ~~Drop~~ ~~remove~~ table employee : also remove data ⇒ ~~Constraint~~ Constraint

Ex How To remove a constraint?

→ Use Alter & drop

Alter table employee drop constraint EmpFK;

TC : ~~القائمة~~ القائمة التي فيها

شؤون الطلاب

* Foreign Key

← استعمل مثال الطالب راكسات

1) يعني في row له id غير موجود في student

2) اذا حذفنا row من student

Both affected ← delete/update ← Cascade A

None is affected ← Reject ← No Action B ON UPDATE NO ACTION

Both affected ← Set Null / Set default C

* Constraints Not Null

Unique : يترك انه جميع columns مختلفة

Default : اعطى قيمة افتراضية

Check : مكان افعلها الراس ابي تبطل اذا صح

ER Diagram To Relational Model

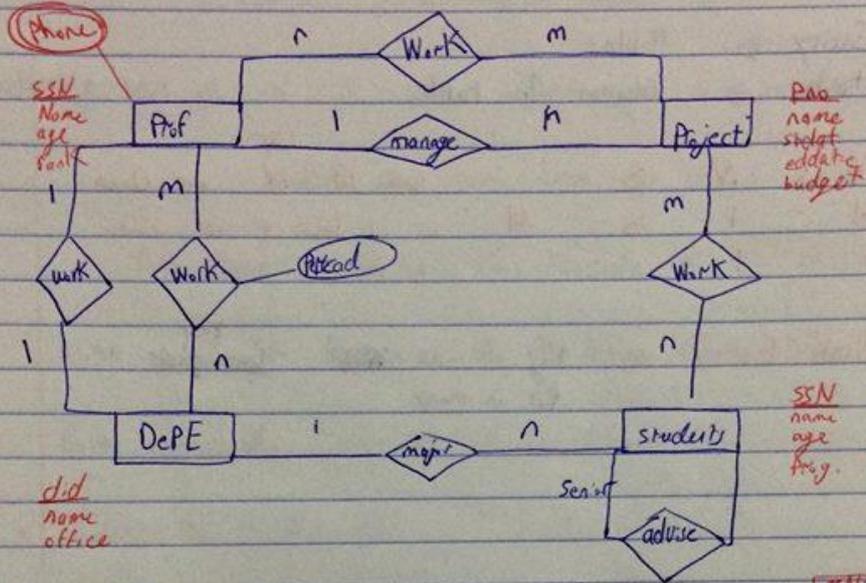
Entity \Rightarrow Tables
Attributes \Rightarrow Column In table

Cardinality Ratio

M x N	\Rightarrow	id as class
M - 1	\Rightarrow	M use 1 key id class
1 - 1		

Weak Entity \Rightarrow Key of one ~~PK~~ ^{Key} ~~PK~~ ^{PK}
Part in many

Ex For the following ERD, find schema?

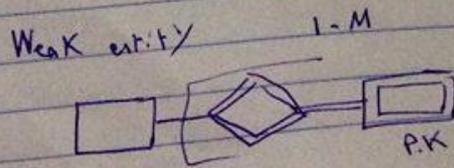


Prof
 SSN: int, name: string, age: int, rank: int

Prof 2 Proj work
 SSN: int

Proj
 Pno: int, name: string, start date: date, end date: date, budget: int

1:1 ⇒ Merge into ball and use



* Schema :-

Find the Schema for Musicians Example

- ① There is a table for each entity
- ② Merging : M-N a new table
 M-1 a Merge to M side (Musicians & Inst)
 1-1)
- ③ ——— not null ← initial null etc

musicians (SSN int, name : string, phone : string)

address (phone : string, street : string, city : string)

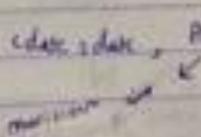
Instrument (inst_id int, name : string, make : string)

mus2Inst (SSN int, inst_id int)

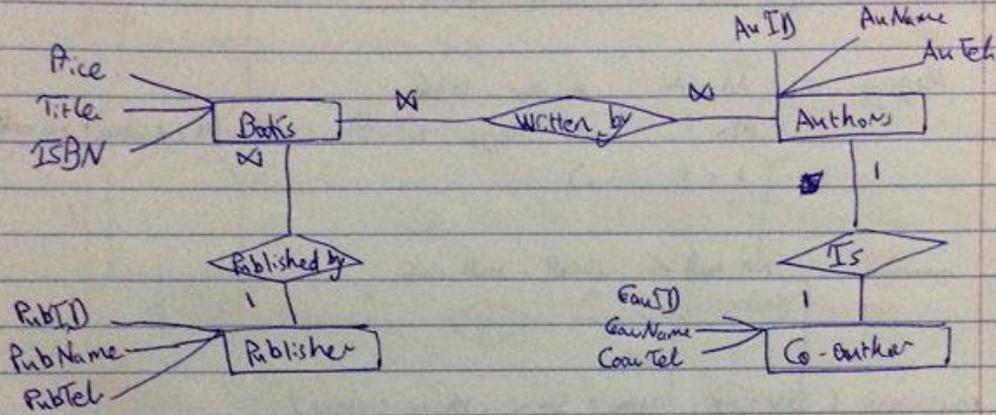
Song (Song_id int, title : string, author : string, Alb_id int)

mus2Song (SSN int, Song_id int)

album (Alb_id int, title : string, cd : date, Prod : int)



- Question
- Ⓐ Write the schema for ERD below
 - Ⓑ Write SQL Statement to create all tables for the system represented by the ERD below



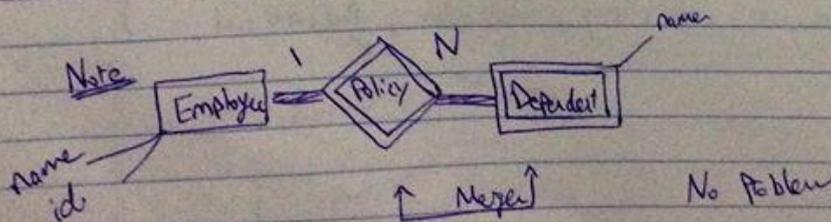
Ans Ⓐ Books (ISBN : string , Price : int , Title : string , PubID : string)

Authors (AuID : int , AuName : string , AuTel : string , CoaID : string)

Book 2 Author (ISBN : string , AuID : int)

Publisher (PubID : string , PubName : string , PubTel : string)

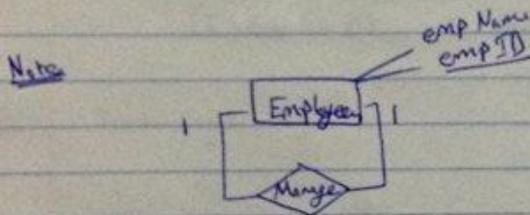
Co-author (CoaID : string , CoaName : string , CoaTel : string)



string $\begin{cases} \rightarrow \text{char} \\ \rightarrow \text{Varchar} \end{cases}$

15

(B) Create Table Books (
ISBN Varchar (20),
.....
.....
.....
PubID Varchar (20),
Primary Key (ISBN),
Foreign Key (PubID) references Publisher (to add it
),



And schema \Rightarrow Employee (empID : int, empName : String, ManagerID : int)

Note Class Hierarchies are only for Hourly Emp & Contract Emp

