

NAME: ZAHOR Khan

B. D. :- 13835

Semster: 7th -

Subject: Data Base System.

Medm: Rimsha Khan.

<u>ID</u>	Student ID	Student Name	address Student.	ID Course	Name Course
01	Fawad	Karachi	SE-01	AI	
01	Fawad	Karachi	SE-05	SDE	
02	Walkeel	Lahore	SE-02	DIID	
03	Saira	Peshawar	SE-03	DB	
03	Saira	Peshawar	SE-04	SRE	
04	Aiman	Karachi	SE-03	DB	
05	Daniyal	Lahore	SE-01	AI	
06	Emaan	Peshawar	SE-01	AI	

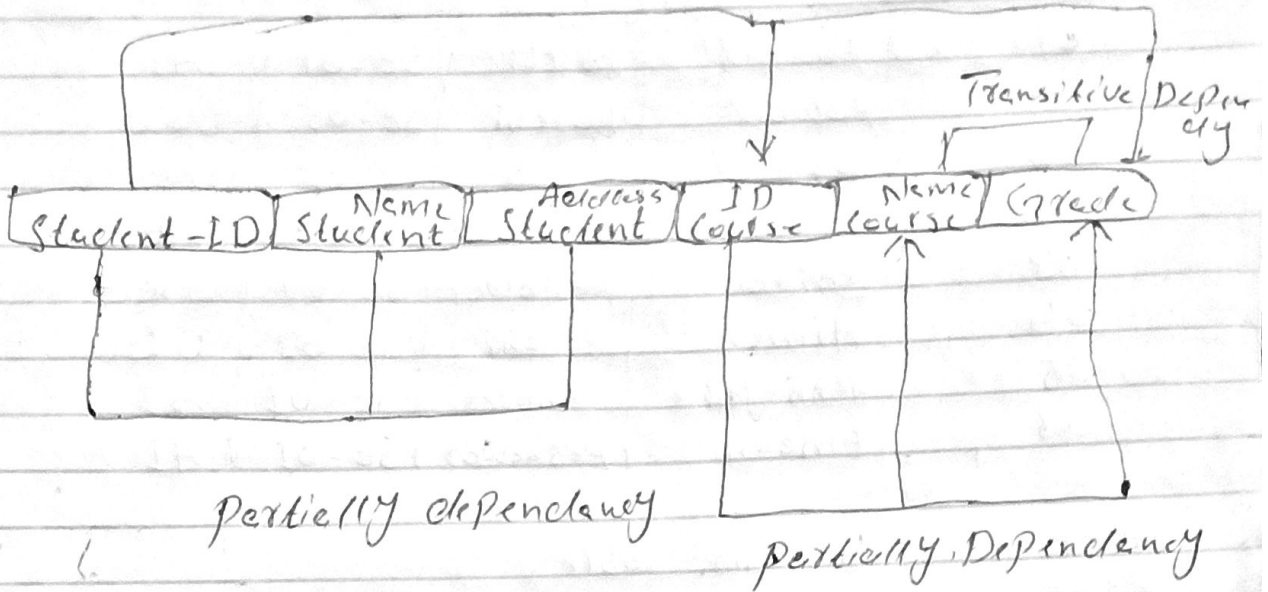
Anomalies in this table

- Insertion - if new a new course of existing student added, course delete must be re-entered.

causing duplication

- Deletion - if we delete the SDE 01, we lose information concerning this item finish and price.
- Update - Changing the details of student ID 01 requires update in several records.

Incl Form



Prime key is student ID  
 Removing Partially Dependency.

Student ID	Course ID	Grade
Student ID	Student Name	Student Address
Course ID	Course Name	Grade

Page (3)

ID: 13835

3rd Normalization Form -

[Student ID] [course ID] [Grade]

[Student ID] [Student Name] [Student Address]

[course ID] [course Name] [Grade]

A diagram showing a box containing three fields: [course ID], [course Name], and [Grade]. A vertical line is drawn under [course ID]. Two arrows point upwards from the bottom of this line to the [course Name] and [Grade] fields, indicating a functional dependency.

Transitive Dependency

[Student ID] [course ID]

[course ID] [course Name] [Grade]



Q2 part (1)

Create table "student" (

ID INT PK-ID Primary Key NOT Null

Student-Name INT (100) NOT Null

DoB Date

Age Integer

Select Max (Age)

From [year]

Where Year (30)

C GPA Float

);

Q2 (2)

Insert Into (Student-ID,

Student-Name, Age, (GPA)

Values (13835, Zahoor Ishaq, 22, 3.5)

Insert into

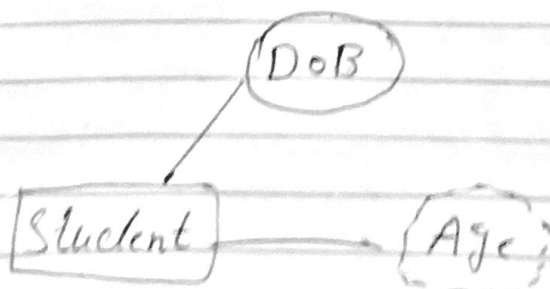
(Student-Id, Student-Name, Age, (GPA

Values (6618), Fahim Ishaq, 25, 3.4

Q3 (3) Derived attributes are those attributes

which can be derived from other attribute(s)

Here, the attribute "Age" is a derived attribute as it can be derived from the attribute "DoB"



Q3 Consider you have the following  
= 2 tables

(1) Write SQL Query for finding & displaying  
Product names and IDs of  
Products  
whose unit price is less than 50

Ans

centeen table

```
SELECT Product-name, Product-ID,  
unit-price
```

```
FROM centeen-table
```

```
WHERE - unit-price < 50
```

For orders:

```
SELECT ORDER-ID, Product-ID, unit  
price
```

```
FROM centeen-table
```

```
WHERE - unit-price < 50
```

(2) Write SQL Query for displaying  
sorted names of product names with  
Alias name as Product-list-sorted

Ans In Ascending: By default, the

sort is performed in ascending order.  
Therefore,

there is no need to specify the  
ASC keyword.

```
SELECT Product Name As
```

```
[Product-Name]
```

```
FROM centeen table
```

(3) Delete data from order-Details  
 whose Quantity is less than 1.

Also delete from order-Details  
 where Quantity is  $\leq 1$ .

(4) Write SQL INNER join query and  
 its output on the given two  
 tables.

Ans SELECT canteen-Table.Product-ID,  
 canteen-Table.Product-Name,  
 order, order-ID, order, order Quantity,  
 order, order-ID, order, order-Quantity  
 FROM canteen-table  
 INNER-Table-Details  
 ON canteen-table, order-ID  
 = order, order-ID;