College of Electrical & Mechanical Engineering, NUST

Department of Computer Engineering

DATABASE ENGINEERING (EC-240)

LAB MANUAL # 06

Course Instructor: Prof Dr Farooque Azam / Assoc Prof Dr Wasi Haider Butt
Lab Engineer: Engineer Kashaf Raheem

Student Name: AMINA QADEER

Degree/ Syndicate: **DE-CE-42-A**

Lab#06: Relational Data Model and Integrity Constraints

Lab Objective:

To introduce students about relational data model and some data integrity constraints that are applied on it.

Lab Description:

Relational data model is a form of logical data model that represents data in the form of tables. A relation is a named, two-dimensional table of data. Each relation (or table) consists of a set of named columns and an arbitrary number of unnamed rows.

EMPLOYEE1

EmpID	Name	DeptName	Salary
100	Margaret Simpson	Marketing	48,000
140	Allen Beeton	Accounting	52,000
110	Chris Lucero	Info Systems	43,000
190	Lorenzo Davis	Finance	55,000
150	Susan Martin	Marketing	42,000

We can express the structure of a relation by using a shorthand notation in which the name of the relation is followed (in parentheses) by the names of the attributes in that relation. For EMPLOYEE1 we would have:

EMPLOYEE1(EmpID, Name, DeptName, Salary)

Relational Keys:

- **Primary Key:** A primary key is an attribute or a combination of attributes that uniquely identifies each row in a relation. We designate a primary key by underlining the attribute name(s).
- **Foreign Key:** A foreign key is an attribute (possibly composite) in a relation that serves as the primary key of another relation.
- **Composite key:** It is a primary key that consists of more than one attribute.

For example, consider the relations EMPLOYEE1 and DEPARTMENT:

EMPLOYEE1(EmpID, Name, DeptName, Salary)
DEPARTMENT(DeptName, Location, Fax)

The foreign key here is DeptName, represented by dashed underline in Employee1 relation.

EMPLOYEE1(EmpID, Name, DeptName, Salary)

Properties of Relations:

We have defined relations as two-dimensional tables of data. However, not all tables are relations. Relations have several properties that distinguish them from non-relational tables that are summarized below:

- Each relation (or table) in a database has a unique name.
- An entry at the intersection of each row and column is atomic (or single valued). There can be only one value associated with each attribute on a specific row of a table; no multivalued attributes are allowed in a relation.
- Each row is unique; no two rows in a relation can be identical.
- Each attribute (or column) within a table has a unique name.
- The sequence of columns (left to right) is insignificant. The order of the columns in a relation can be changed without changing the meaning or use of the relation.
- The sequence of rows (top to bottom) is insignificant. As with columns, the order of the rows of a relation may be changed or stored in any sequence.

Integrity Constraints:

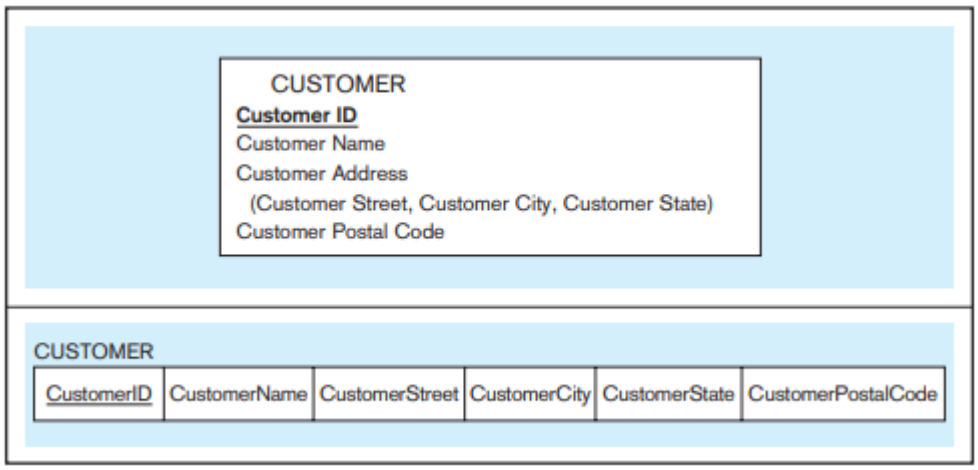
The relational data model includes several types of constraints, or rules limiting acceptable values and actions, whose purpose is to facilitate maintaining the accuracy and integrity of data in the database. The major types of integrity constraints are domain constraints, entity integrity, and referential integrity.

- **Domain Constraints:** All of the values that appear in a column of a relation must be from the same domain. A domain is the set of values that may be assigned to an attribute.
- **Entity Integrity:** The entity integrity rule is designed to ensure that every relation has a primary key and that the data values for that primary key are all valid. In particular, it guarantees that every primary key attribute is non-null.

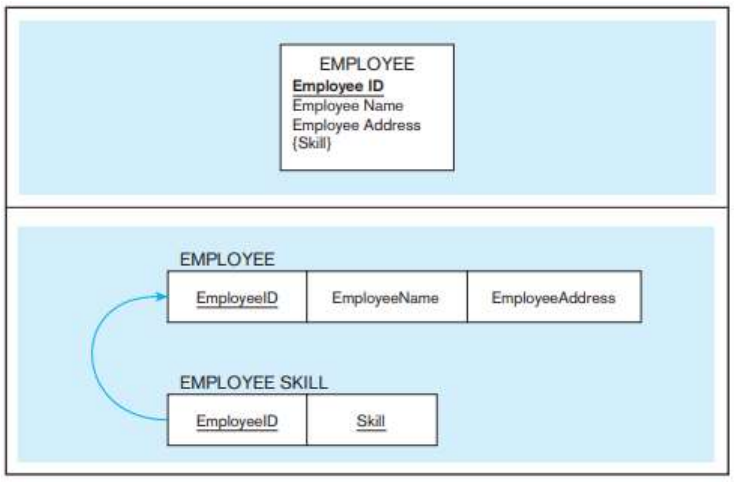
- **Referential Integrity Constraint:** It is a rule that maintains consistency among the rows of two relations. The rule states that if there is a foreign key in one relation, either each foreign key value must match a primary key value in another relation or the foreign key value must be null.

Mapping Entities to Relations:

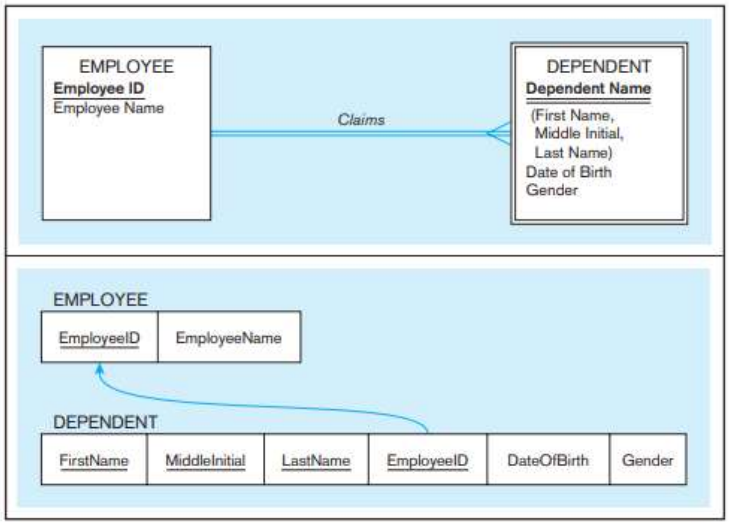
- **Mapping Composite Attributes:** When a regular entity type has a composite attribute, only the simple components of the composite attribute are included in the new relation as its attributes.



- **Mapping Multi-valued Attributes:** When the regular entity type contains a multivalued attribute, two new relations (rather than one) are created. The first relation contains all of the attributes of the entity type except the multivalued attribute. The second relation contains two attributes that form the primary key of the second relation. The first of these attributes is the primary key from the first relation, which becomes a foreign key in the second relation. The second is the multivalued attribute.



Mapping Weak Entities: For each weak entity type, create a new relation and include all of the simple attributes (or simple components of composite attributes) as attributes of this relation. Then include the primary key of the identifying relation as a foreign key attribute in this new relation. The primary key of the new relation is the combination of this primary key of the identifying and the partial identifier of the weak entity type.



Lab Tasks:

01. Consider the following entities and attributes of a pine valley furniture case description.

CUSTOMER (CustomerID, CustomerName, CustomerAddress, CustomerCity, CustomerState, CustomerPostalCode)
ORDER (OrderID, OrderDate, CustomerID)
ORDER LINE (OrderID, ProductID, OrderedQuantity)
PRODUCT (ProductID, ProductDescription, ProductFinish, ProductStandardPrice, ProductLineID)

To Dos:

- Convert the entities with attribute in relations.
- Specify Domain Constraints i.e., shows domain definitions for the domains associated with the attributes.
- Show Entity Integrity i.e., no primary key attribute (or component of a primary key attribute) may be null.
- Identify Foreign key were applied and show referential integrity i.e., foreign key value must match a primary key value in another relation, or the foreign key value must be null.

Solution:

Domain Constraints:

CustomerID: alphanumeric, unique
CustomerName: string
CustomerAddress: string
CustomerCity: string
CustomerState: string
CustomerPostalCode: alphanumeric

Entity Integrity:

CustomerID: not null
ORDER (OrderID (PK), OrderDate, CustomerID (FK))

Domain Constraints:

OrderID: alphanumeric, unique
OrderDate: date

CustomerID: alphanumeric

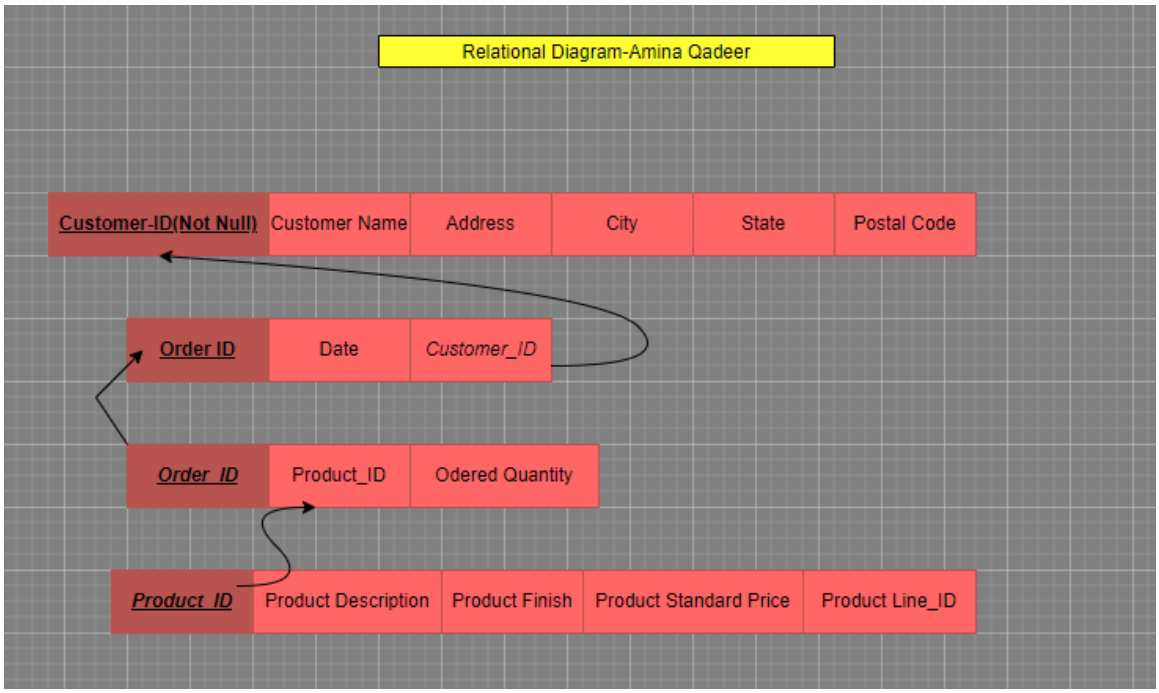
Entity Integrity:

OrderID: not null

CustomerID: not null

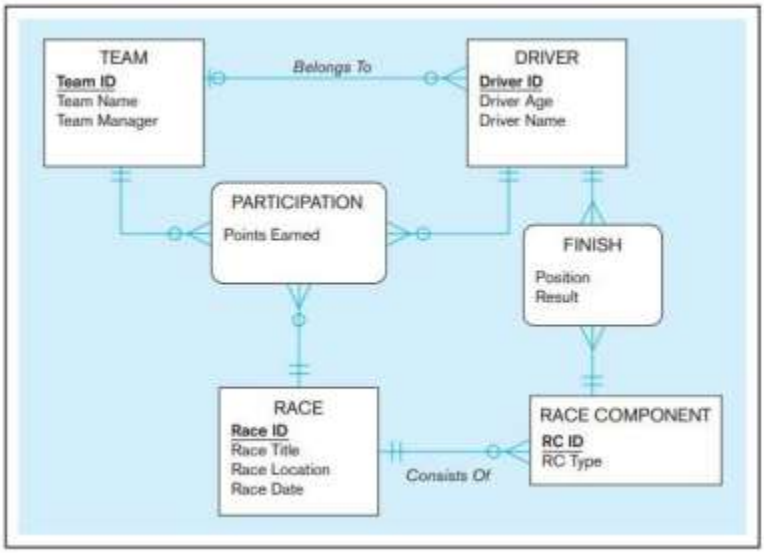
Foreign Key:

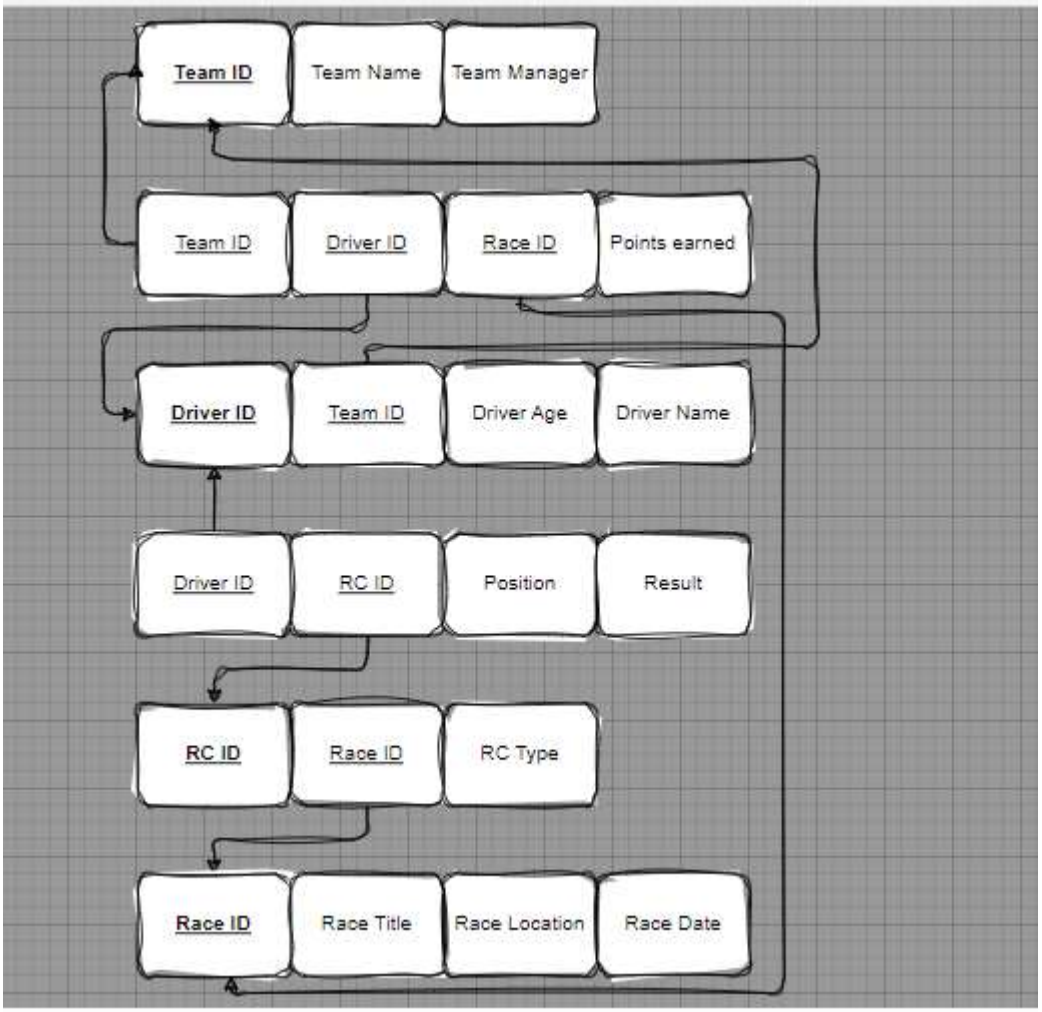
CustomerID references CUSTOMER(CustomerID)



Arrow head from primary key towards foreign key* head pointing to PK and tail pointing to FK

02. Figure includes an EER diagram describing a car racing league. Transform the diagram into a relational schema that shows referential integrity constraints.

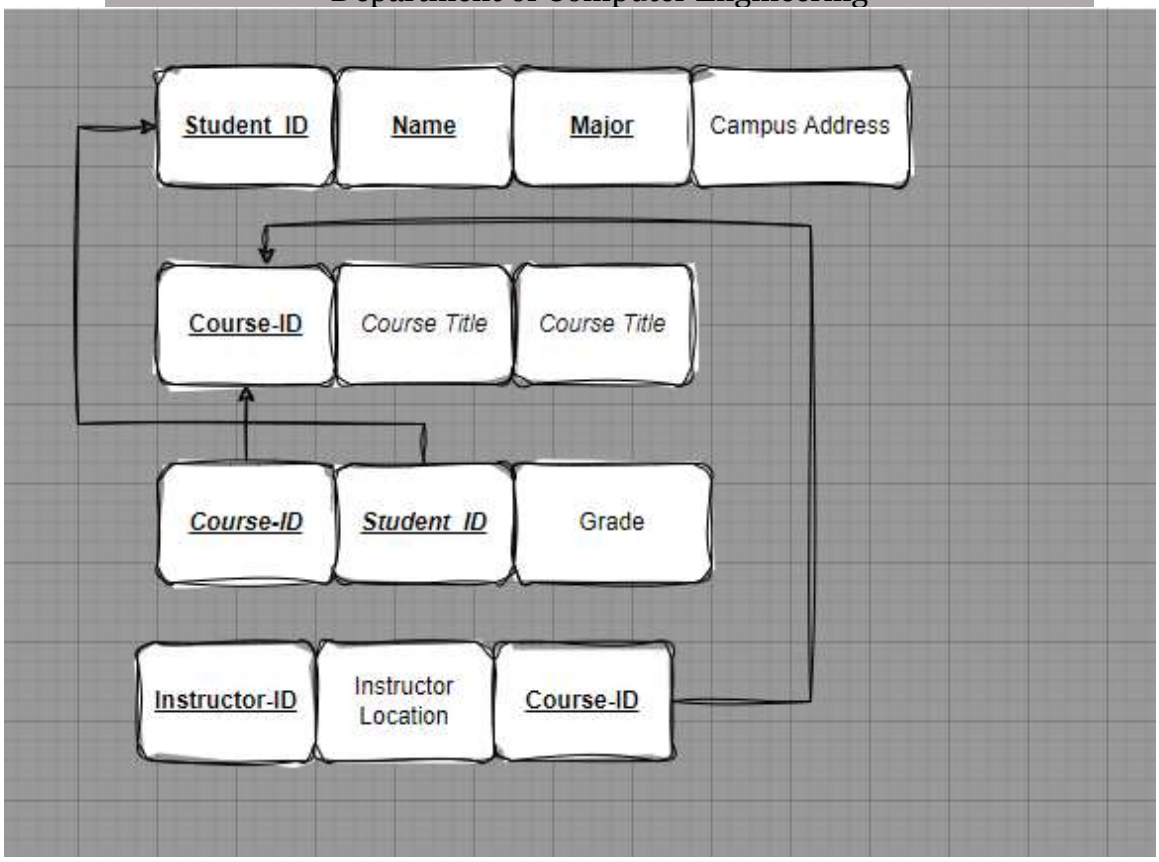




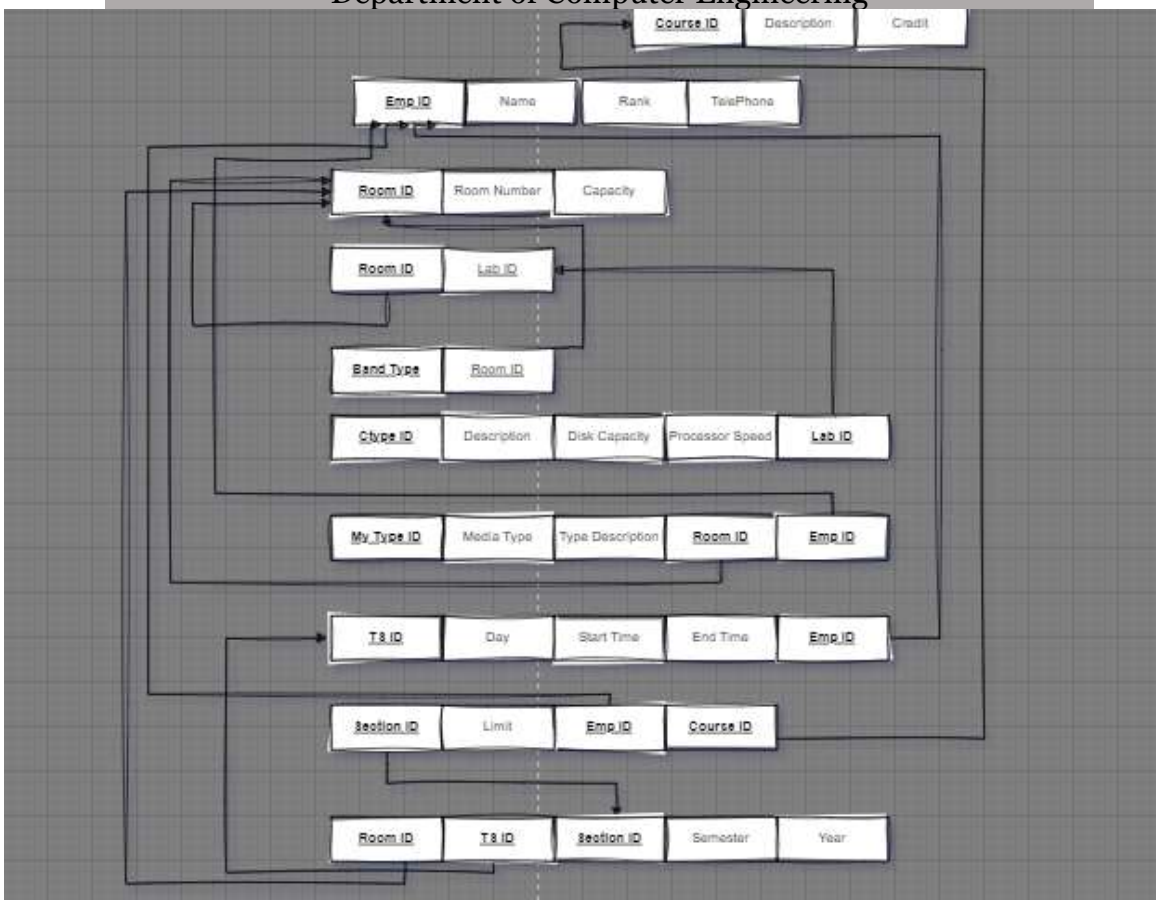
03. Table shows a relation called GRADE REPORT for a university. Draw a relational schema.

Grade Report

StudentID	StudentName	CampusAddress	Major	CourseID	CourseTitle	Instructor Name	Instructor Location	Grade
168300458	Williams	208 Brooks	IS	IS 350	Database Mgt	Codd	B 104	A
168300458	Williams	208 Brooks	IS	IS 465	Systems Analysis	Parsons	B 317	B
543291073	Baker	104 Phillips	Acctg	IS 350	Database Mgt	Codd	B 104	C
543291073	Baker	104 Phillips	Acctg	Acct 201	Fund Acctg	Miller	H 310	B
543291073	Baker	104 Phillips	Acctg	Mkgt 300	Intro Mktg	Bennett	B 212	A



04. Convert the EERD of lab 05 Task 3 into relational model.


Course ID
Description
Credit
Emp ID
Name
Rank
TelePhone
Room ID
Room Number
Capacity
Room ID
Lab ID
Band Type
Room ID
Ctype ID
Description
Disk Capacity
Processor Speed
Lab ID
My Type ID
Media Type
Type Description
Room ID
Emp ID
T8 ID
Day
Start Time
End Time
Emp ID
Section ID
Limit
Emp ID
Course ID
Room ID
T8 ID
Section ID
Semester
Year