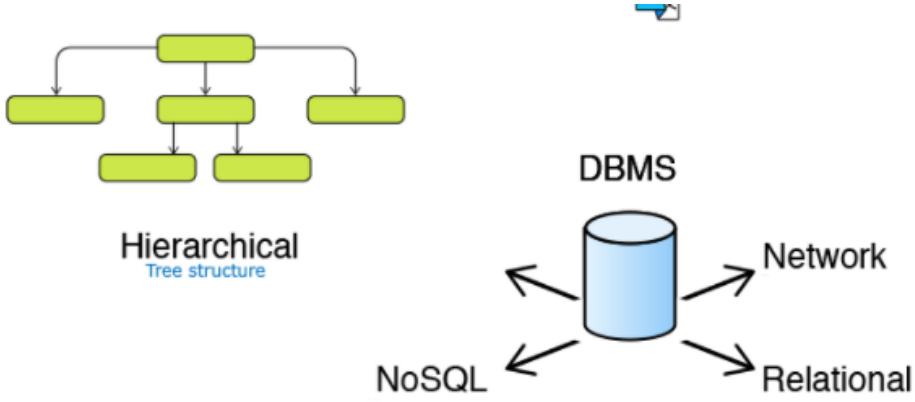
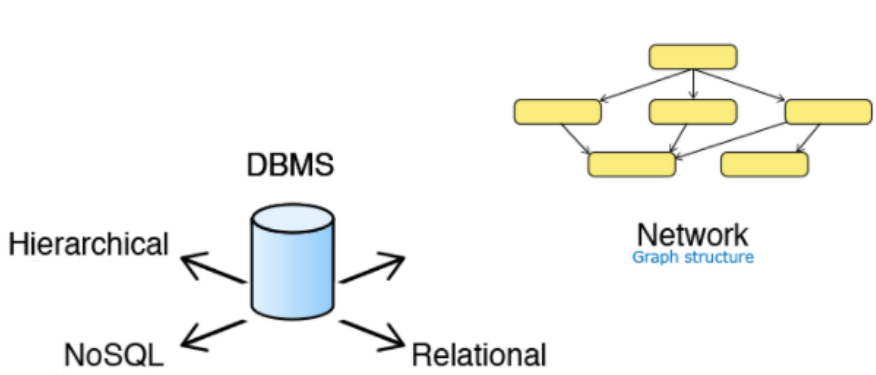
Types of Database Systems

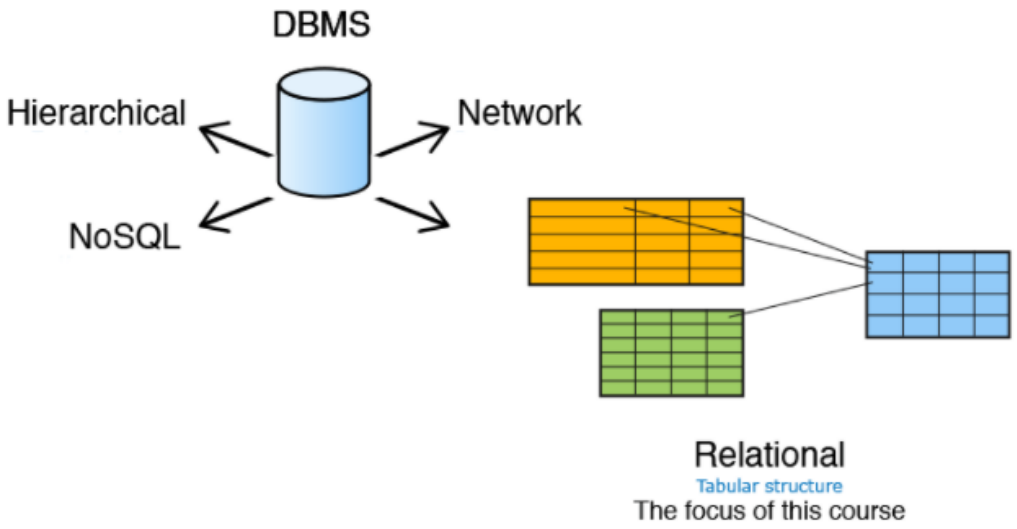
Hierarchical Databases organize data into a tree-like structure. Data is stored as records which are connected to one another through parent child relationships. Some examples of Hierarchical Databases are Information Management System (IMS), Raima Database Manager (RDM) Mobile etc.



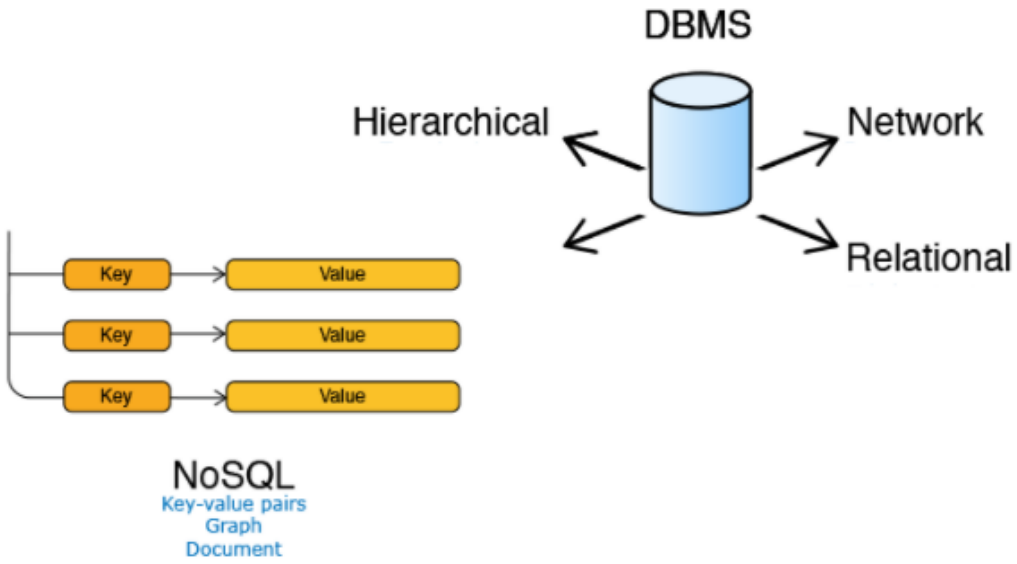
Network Databases organize data into a graph structure in which object types are nodes and relationship types are arcs. Each record can have multiple parent and child records. Some examples of Network Databases are Integrated Database Management System (IDMS), Integrated Data Store (IDS) etc.



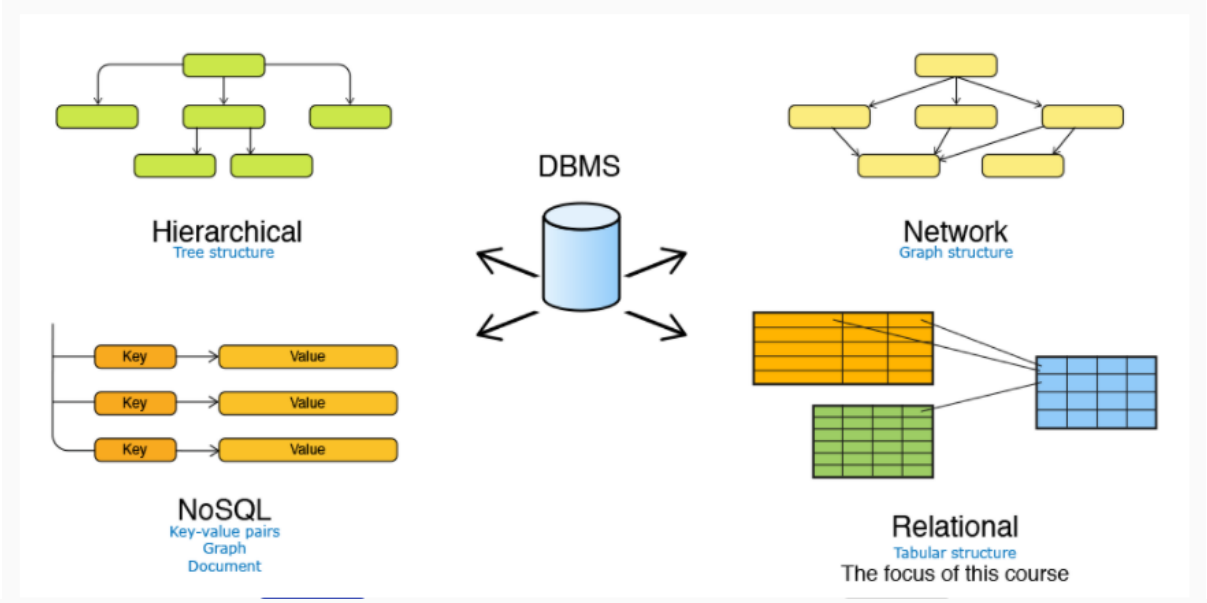
Relational Databases organizes data into one or more tables. A table consists of attributes (columns), tuples (rows) and provides a way to uniquely identify each tuple. Tables are related to each other through parent child relationships. Some examples of Relational Databases are DB2, Oracle, SQL Server etc.



NoSQL (Not only SQL) database uses key-value, graph or document data structures to store data. These databases aim for simplicity of design, horizontal scaling and finer control over availability. Some examples on No Sql databases are Cassandra, MongoDB, CouchDB, OrientDB, HBASE etc.



Additionally there are other databases types as well like Object Oriented databases e.g. DB4O and ZopeDB, Graph Databases e.g. Neo4J and InfiniteGraph etc. Relational databases are the most widely used database in the current times. In this course we will focus on Relational Databases.



- An ***attribute*** is a named column of a relation. It stores a specific information about an object e.g. salary.

Attributes / columns / fields

ID	ENAME	SALARY	BONUS	DEPT
1	James Potter	75000	1000	ICP
2	Ethan McCarty	90000	2000	ETA
3	Emily Rayner	25000		ETA
4	Jack Abraham	30000	1000	ETA

- A ***tuple*** is a row in a relation. It represents relationship between attributes that can contain single value.

Rows / records / tuples

ID	ENAME	SALARY	BONUS	DEPT
1	James Potter	75000	1000	ICP
2	Ethan McCarty	90000	2000	ETA
3	Emily Rayner	25000		ETA
4	Jack Abraham	30000	1000	ETA

- ***Cardinality*** of relation is the number of rows it contains. e.g. Cardinality of relation below is 4.

ID	ENAME	SALARY	BONUS	DEPT
1	James Potter	75000	1000	ICP
2	Ethan McCarty	90000	2000	ETA
3	Emily Rayner	25000		ETA
4	Jack Abraham	30000	1000	ETA

No of records / rows / tuples:
Cardinality of the relation

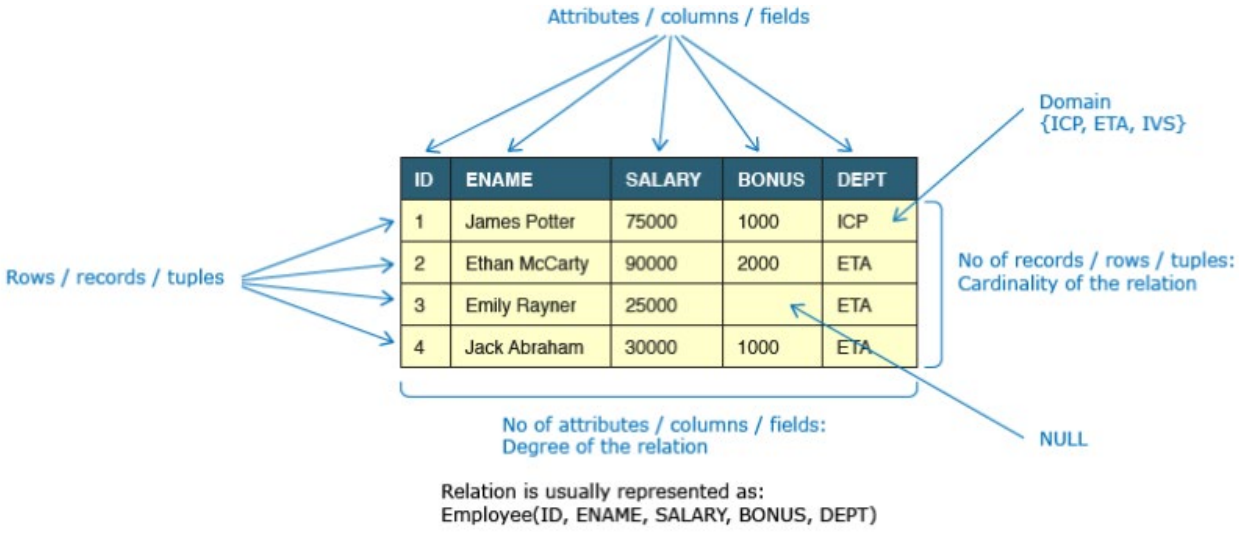
- ***Degree*** of relation is the number of attributes it contains. e.g. Degree of relation below is 5.

ID	ENAME	SALARY	BONUS	DEPT
1	James Potter	75000	1000	ICP
2	Ethan McCarty	90000	2000	ETA
3	Emily Rayner	25000		ETA
4	Jack Abraham	30000	1000	ETA

No of attributes / columns / fields:
Degree of the relation

- A ***domain*** is the set of allowable values for one or more attributes.

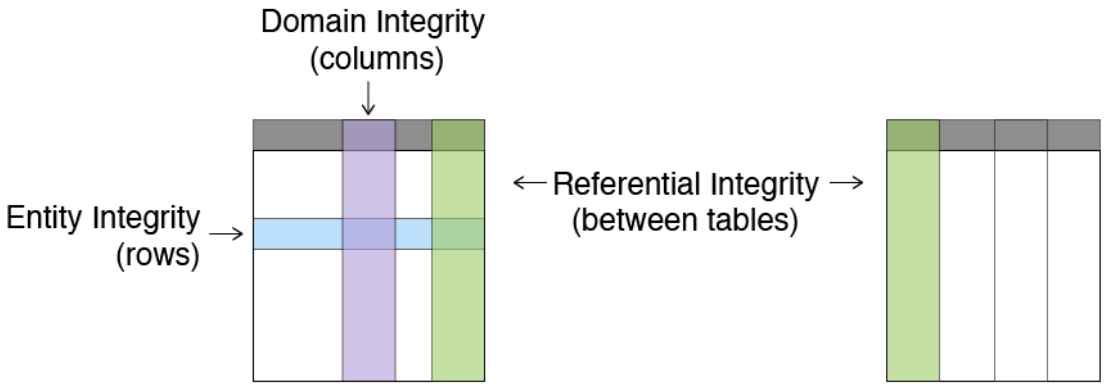
- A collection of relations with distinct relation names is called as Relational Model.



Data Integrity and Constraints

Data integrity refers to maintaining and assuring the accuracy and consistency of data over its entire life-cycle. Database Systems ensure data integrity through constraints which are used to restrict data that can be entered or modified in the database. Database Systems offer three types of integrity constraints:

Integrity Types	Definition	Enforced Through
Entity Integrity	Each table must have a column or a set of columns through which we can uniquely identify a row. These column(s) cannot have empty (null) values.	PRIMARY KEY
Domain Integrity	All attributes in a table must have a defined domain i.e. a finite set of values which have to be used. When we assign a data type to a column we limit the values that it can contain. In addition we can also have value restriction as per business rules e.g. Gender must be M or F.	DATA TYPES, CHECK CONSTRAINT
Referential Integrity	Every value of a column in a table must exist as a value of another column in a different (or the same) table.	FOREIGN KEY



- A **Candidate Key** is a minimal set of columns/attributes that can be used to uniquely identify a single row/tuple in a relation.
- A **Primary Key** is the candidate key that is selected to uniquely identify a tuple in a relation. The mandatory and desired attributes for a primary key are:

Mandatory	Desired
must uniquely identify a tuple	should not change with time
must not allow NULL values	should have short size e.g. numeric data types

- A ***Foreign Key*** is a set of one or more columns in the child table whose values are required to match with corresponding columns in the parent table. Foreign key establishes a relationship between these two tables. Foreign key columns identified in child tables must refer to the primary key or unique of the parent table. The child table can contain NULL

values. Let us take an example of Employee and Computer tables as provided below:

Parent / master / referenced table

Computer table

COMPID	MAKE	MODEL
1001	Dell	Vostro
1002	Dell	Precision
1003	Lenovo	Edge

Child / referencing table

Employee table

ID	ENAME	DEPT	COMPID
1	James Potter	ICP	1001
2	Ethan McCarty	ETA	NULL
3	Emily Rayner	ETA	1002

Computer is the parent table with COMPID as the primary key. Employee is the child table with ID as the primary key. If we want to allocate a maximum of one computer to an employee then COMPID must be made the foreign key in the Employee table. It can only contain values that are present in Computer table COMPID column or no values at all(NULL). We cannot allocate a computer that does not exist to an employee.

Additionally, multiple rows in the child table can link to the same row of the parent table depending upon the type of relationship.

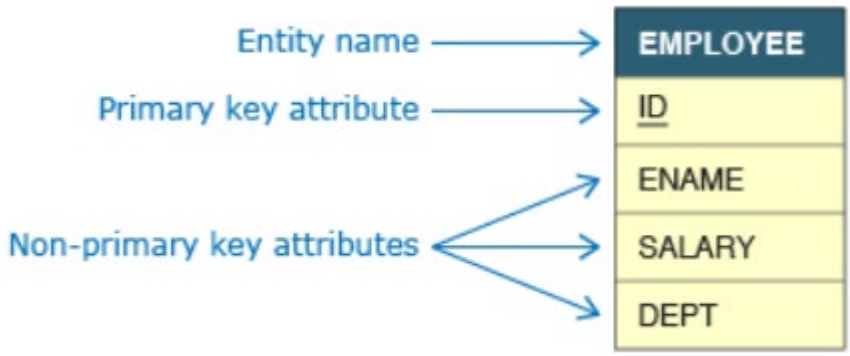
➢ Entity and Relationships

Jack is a part of the database team and he needs to present the database design to business users. The business users are non-technical and it is difficult for them to read a verbose design document. What can Jack do? Jack needs to use an Entity Relation (ER) Model.

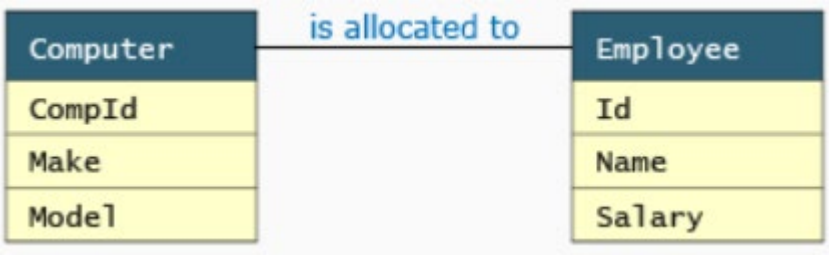
➢ ***ER model*** is a graphical representation of entities and their relationships which helps in understanding data independent of the actual database implementation. Let us understand some key terms used in ER Modelling.

Term	Definition	Examples
Entity	Real world objects which have an independent existence and about which we intend to collect data.	Employee, Computer
Attribute	A property that describes an entity.	Name, Salary

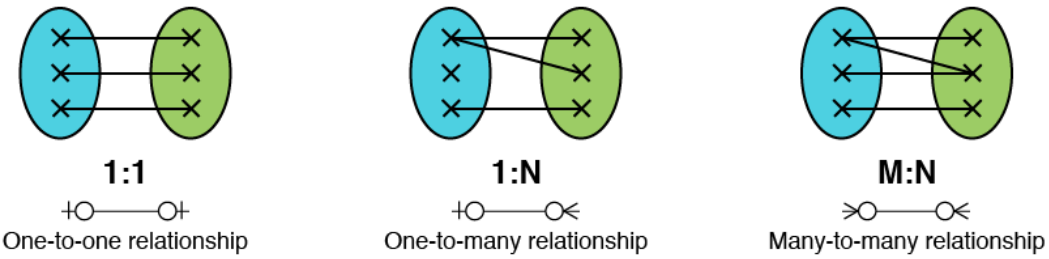
A sample ER Diagram representing the Employee entity along with its attributes is presented below:



- ***Relationships*** are associations of one entity with another entity through a foreign key. Each relationship has a name e.g. a Computer is allocated to an Employee.

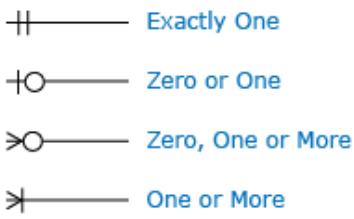


- ***Cardinality of relationship*** is the number of instances in one entity which is associated to the number of instances in another. For the relationship between Employee and Computer, it helps us answer questions like how many computers can be allocated to an employee, can computers be shared between employees, can employees exist without being allocated a computer etc. e.g. if 0 or 1 computer can be allocated to 0 or 1 employee then the cardinality of relationship between these two entities will be 1:1.
- Cardinality of relationships are of three types: 1:1, 1:N and M:N.

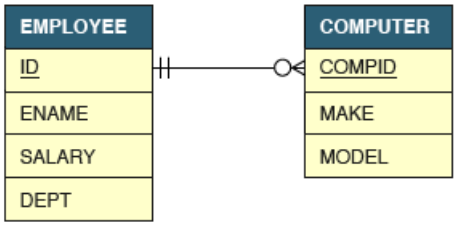


A relationship with cardinality 1:1 is also called as one-to-one relationship or 1:1 relationship.

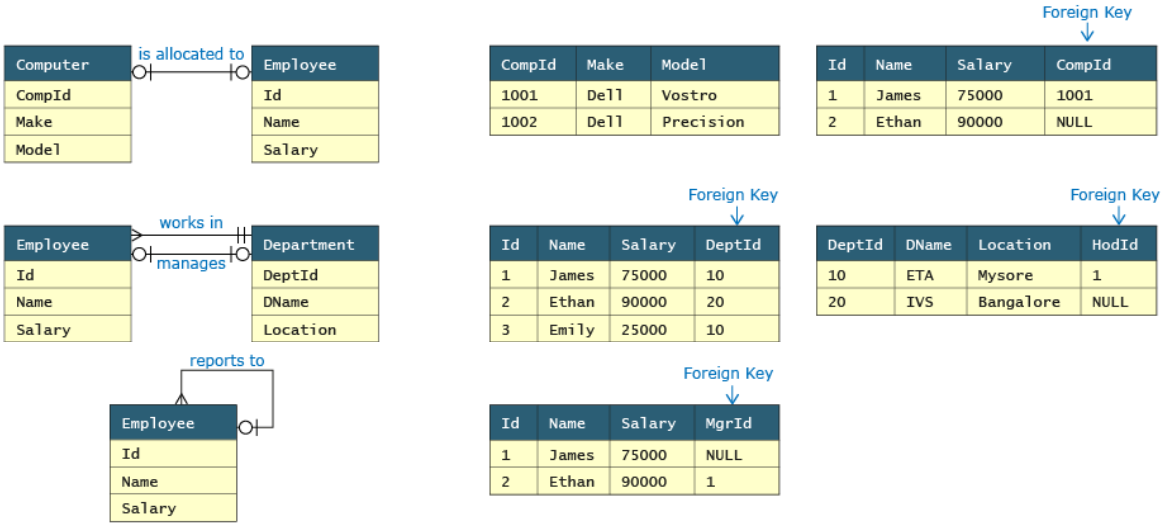
- **Crow foot notation** is one of the ways to represent cardinality of relationship in an ER Model. The notation comprises of four symbols and one of them needs to be used for each entity in a relationship.



Let us say the relationship between employee and computer is such that a computer must be allocated to one and only one employee but an employee can be allocated with zero or any number of computers. Such a relationship is represented by the diagram below:



Foreign keys need to be created in tables in order to establish the relationship between entities.



The table in which foreign key will be created depends upon the cardinality of the relationship. Let us now discuss about the types of cardinalities and how it impacts foreign key creation.

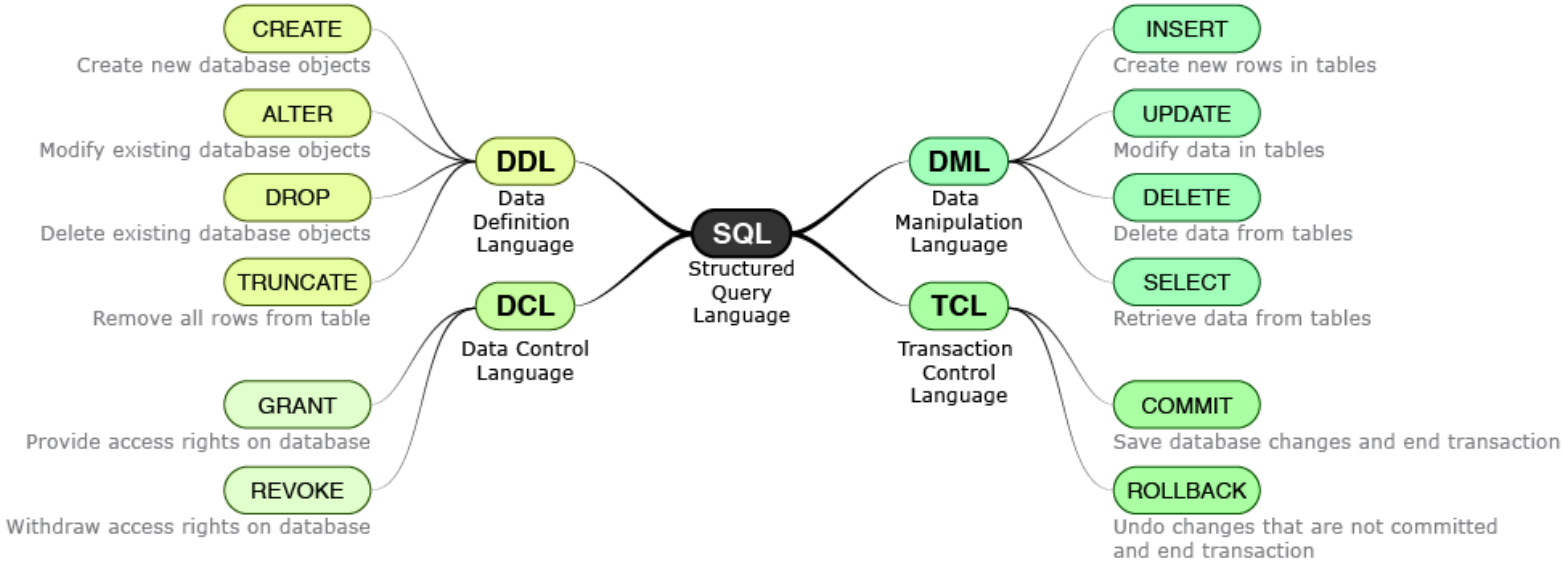
➢ Difference between Primary Key(PK) and Unique Key(UK)

	PRIMARY KEY (PK)	UNIQUE KEY (UK)
Allows only unique values?	Yes	Yes
Composite keys allowed?	Yes	Yes
NULL values allowed?	No	Yes
Multiple keys allowed?	Zero or one PK per table	Zero or more UK per table

SQL BASICS

SQL Commands and Data Types

Structured Query Language (SQL) is used to manage data in all relational databases like DB2, Oracle, SQL Server etc. SQL standards are maintained by ISO. While most database products comply with the ISO standard, they also offer additional proprietary features. In this course we will restrict ourselves to feature set offered by Oracle database.



- ***Data Definition Language*** is used to specify the structure i.e. schema of a relational database. DDL provides commands for creation, modification and deletion of various database objects like tables, views, stored procedures, indexes, constraints etc. The output of DDL is placed in data dictionary which contains metadata i.e. data about data.
- ***Data Manipulation Language*** enables users to access or manipulate data in a relational database. DML provides commands for retrieval, creation, deletion and modification of information in a database. DML requires a user to specify what data is needed without specifying how to get it. The database engine is left to figure out effective means of retrieving data.
- ***Data Control Language*** enables users to provide access to various database objects like views, tables, stored procedures etc. in a relational database. Typically only DBAs have access to grant and revoke privileges. Whenever a user submits a query, the database checks against the granted privileges and rejects the query if it is not authorized.
- ***Transaction Control Language*** specifies commands for beginning and ending a transaction. A transaction consists of a sequence of SQL statements that are applied in an atomic (all or none) manner. A commit makes all the changes applied by the transaction permanent on the database while a rollback undoes all the changed applied by the transaction.

SQL Data Types :

Data types :

- **Character data types**
- **Integral data types**
- **Non-Integral data types**
- **Miscellaneous data types**

Operators :

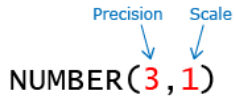
- **Arithmetic operators**
- **Comparison operators**
- **Logical operators**

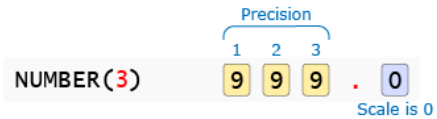
	CHAR(n)	VARCHAR2(n)
Useful for	Storing characters having pre-determined length	Storing characters whose length vary a lot
Storage size	size for n characters	size for actual no. of characters + fixed size to store length
Storage Characteristic	Trailing spaces are applied if data to be stored has smaller length than n.	Trailing spaces are not applied.
Maximum size	2000 bytes	4000 bytes
Example	A CHAR(10) field will store "Hello" as 10 bytes by appending 5 trailing spaces.	A VARCHAR2(10) field will store "Hello" as 7 bytes (assuming 2 bytes to store length).
Alternate Name	CHARACTER(n)	CHARACTER VARYING(n)

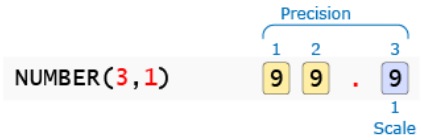
- SQL supports SMALLINT, INTEGER and INT data types that are used for storing whole numbers. Unlike other databases, Oracle does not define different size limits for them. They are all treated internally to have 38 digits of precision. Some real-life examples of values are provided below:

Example	Value
Height of Mount Everest in meters	8848
Length of Great Wall of China in meters	885000
Average distance of Earth from the Sun in meters	150000000000

- SQL, unlike programming languages, does not provide support for arbitrary length numbers i.e. numbers not bound by size limits. For e.g. Python supports bignum and Java supports BigInteger data types.
- Nonintegral data types have an integer part and a fractional part. Either NUMERIC, DECIMAL or NUMBER data types can be used to store nonintegral numbers.
- **Scale** is the number of digits allowed after the decimal point. **Precision** is the total number of significant digits i.e. digits both before and after the decimal point. If Scale is not provided then NUMBER datatype can be used to store integral values.







Data Type	Useful for
DATE	Storing date data where time portion is not required. For e.g. Date of Birth, Date of Joining a Company etc. The default format in which date needs to be specified is DD-MON-YY.
TIMESTAMP	Storing date data with precision up-to 1 billionth (9 digits) of a second. Timestamps are typically used as audit fields in database to record the exact time when a transaction occurred.
CLOB (Character Large Object)	Storing large character based data which cannot be stored in VARCHAR2 due to its 4000 bytes size limit.
BLOB (Binary Large Object)	Storing large binary data like movies, images with size up to 4GB.

Here are some key events and their dates from history:

Data Type	Example	Value
DATE	Date Infosys was founded	02-JUL-81
TIMESTAMP	Date Apollo 11 landed on the moon	20-JUL-69 08:18:00.000000 PM

❖ Operators and Expressions

Arithmetic Operators

Operator	Symbol	Usage	Result
Addition	+	15 + 5	20
Subtraction	-	15 - 5	10
Multiplication	*	15 * 5	75
Division	/	15 / 5	3

Comparison Operators

Operator	Symbol	Usage	Result
Equal to	=	15 = 5	false
Not equal to	<>	15 <> 5	true
Greater than	>	15 > 5	true
Greater than equal to	>=	15 >= 5	true
Less than	<	15 < 5	false
Less than equal to	<=	15 <= 5	false

There is one important difference between Equal To comparison operator in programming languages and SQL. While SQL uses a single '=', programming languages typically use double '=' to distinguish it from the assignment operator.

Operation	Python Operator	SQL Operator
Assignment	=	=
Equality check	==	=

Note - SQL also supports operators similar to those in programming languages

- An arithmetic operation involving a NULL returns NULL
- The python operator specified here is applicable for other programming languages like Java, C, C++ etc

Other Comparison Operators

Operator	Symbol	Usage	Example
Range	BETWEEN *<lower limit>* AND *<upper limit>*	Matches value between a range of values (Both inclusive)	Salary BETWEEN 2500 AND 3000
List	IN (List of values)	Matches any of a list of values	Dept IN ('IVS', 'ETA', 'ICP')
String pattern matching	LIKE	Matches a character pattern	SupplierId LIKE 'S%'
NULL Test	IS NULL	Is a null value	Bonus IS NULL

Logical Operators

Operator	Symbol	Usage	Example
And	AND	Returns TRUE if both conditions are true	Salary >= 30000 AND Dept = 'ETA'
Or	OR	Returns TRUE if any one of the condition is true	Salary > 75000 OR Dept = 'ICP'
Not	NOT	Returns TRUE if following condition is false	Id NOT IN (2,3)

- Similar to arithmetic expressions in programming languages, SQL expressions are created from constant values, operators and brackets. They evaluate to a single value and are used in SELECT and WHERE clauses. Some examples are provided below:

Create and Drop Table Syntax

- By now we have understood the different commands, data types and operators of SQL. Let us now see these concepts in action and understand, how it can be used in order to implement different requirements with the help of DDL and DML commands.
- **CREATE TABLE** statement is used to create a table in a database. Database tables are organized into rows and columns. Each table must have a name and can have any number of columns (minimum 1 column is required). Each column must have a data type which determines the type of values that can be stored. CREATE TABLE command will fail if a table is already existing in the database with the same name. All tables must have a unique name.

➢ **DROP TABLE** statement is used to remove an existing table from the database.

DROP TABLE statement is used to drop an existing table from the database.

The syntax for DROP statement:

DROP TABLE <Table name>;

Note:- Tables without any reference with the child table can be dropped using the DROP statement given above.

How to DROP a table?

The steps for the tryout are given below. You can try these statements in the Database playground.

Consider the tables Student and Marks that were seen in the example for explaining Foreign Key.

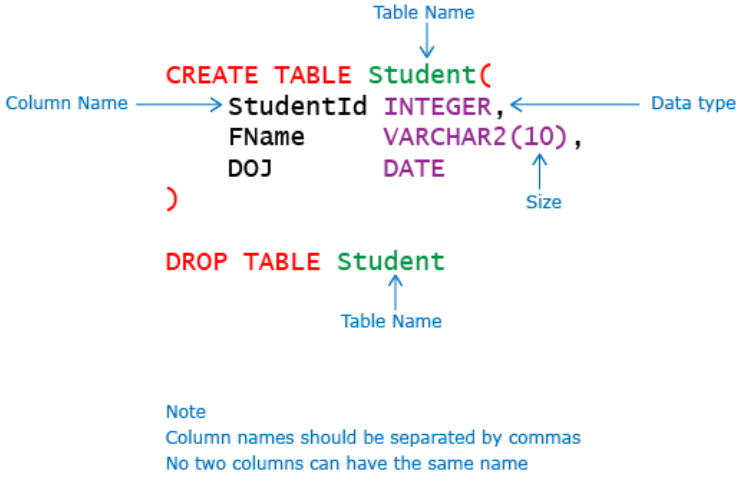
Step 1: Execute the CREATE statement for Student (parent table) as given below:

```
1. CREATE TABLE Student (
2.     Studentid INTEGER CONSTRAINT stud_sid_pk PRIMARY KEY,
3.     FName VARCHAR2(10),
4.     ContactNo NUMBER (10));
5.
```

Step 2: Insert the following records into the table Student.

STUDENTID	FNAME	CONTACTNO
1001	John	8754212356
1002	Jack	7456878956

Step 3: Execute the CREATE statement for Marks (child table) as given below:



```
1. CREATE TABLE Marks (
2.     CourseId INTEGER,
3.     StudentId INTEGER CONSTRAINT marks_sid_fk REFERENCES
   Student (StudentId),
4.     MarksScored DECIMAL (5,2));

5.
```

Step 4: Insert the following records into the table Marks.

COURSEID	STUDENTID	MARKSSCORED
801	1001	79
802	1002	90.75

Step 5: Execute the DROP statement for Student (parent table) as given below:

```
1. DROP TABLE Student;
```

This gives an error as follows:

ORA-02449: unique/primary keys in table referenced by foreign keys

Reason for the error:

The values of the primary/unique key column(s) from the parent table are referred to in the foreign key column of the child table (referential integrity). This reference will not allow the master table to be dropped prior to the child table.

The primary key (Studentid) values of the parent table (Student) 1001 and 1002 are referred to in the foreign key (Studentid) of the child table (Marks). This is **referential** integrity. Hence table Student cannot be dropped prior to table Marks.

Step 6: Execute the DROP statement for Marks (child table) as given below:

```
1. DROP TABLE Marks;
```

Step 7: Execute the DROP statement for Student (parent table) as given below:

```
1. DROP TABLE Student;
```

Both the tables will be dropped successfully from the database.

Reason: When the child table is dropped first, the values that are referred to are removed. So, now the parent table can be dropped from the database without any error.

Conclusion: Drop all the child tables first, then drop all the parent tables.

An alternate method to drop table is given below:

DROP TABLE <Table name> CASCADE CONSTRAINTS;

CASCADE CONSTRAINTS clause should be added to the DROP statement to drop all the referential integrity constraints that refer to primary and unique keys in the table.

The steps for the tryout are given below. You can try these statements in the Database playground.

Step 1: Execute the CREATE statement for Student (parent table) as given below:

```
1. CREATE TABLE Student (
2.     Studentid INTEGER CONSTRAINT stud_sid_pk PRIMARY KEY,
3.     FName VARCHAR2(10),
4.     ContactNo NUMBER (10));

5.
```

Step 2: Insert the following records into the table Student.

COURSEID	STUDENTID	MARKSSCORED
801	1001	79
802	1002	90.75

Step 3: Execute the CREATE statement for Marks (child table) as given below:

```
1.     CREATE TABLE Marks (
2.     CourseId INTEGER,
3.     StudentId INTEGER CONSTRAINT marks_sid_fk REFERENCES
  Student (StudentId),
4.     MarksScored DECIMAL (5,2));

5.
```

Step 4: Insert the following records into the table Marks.

COURSEID	STUDENTID	MARKSSCORED
801	1001	79
802	1002	90.75

Step 5: Execute the DROP statement for Student (parent table) as given below:

```
1. DROP TABLE Student CASCADE CONSTRAINTS;
```

Step 6: Execute the DROP statement for Marks (child table) as given below:

```
1. DROP TABLE Marks CASCADE CONSTRAINTS;
```

Both the tables will be dropped successfully from the database.

Reason: Since the referential integrity constraint is dropped from the parent table, the parent table can be dropped first, followed by the child table.

Constraint Type	Applies On
Single Column Constraint	Single Column
Composite Constraint	Multiple columns

Constraint Type	Specified
Column Level Constraint	With Column definition
Table Level Constraint	After Column definition

The primary key (Studentid) values of the parent table (Student) 1001 and 1002 are referred to in the foreign key (Studentid) of the child table (Marks). The clause CASCADE CONSTRAINTS will remove all referential integrity constraints that refer to the primary or unique keys in the table.

The referential integrity constraints that refer to the primary key (Studentid) values of the parent table (Student) 1001 and 1002 are removed. So, now the parent table (Student) can be dropped before the child table (Marks).

Conclusion: CASCADE CONSTRAINTS clause will drop all the referential integrity constraints that refer to the primary or unique keys in the table. Hence all the parent tables can be dropped before the child tables.

❖ Constraints :

We have learnt that data integrity in database systems is enforced through constraints. These constraints are typically specified along with the CREATE TABLE statement. Constraints are classified into multiple types based on the number of columns they act upon as well as on the way they are specified.

Table level constraint can be specified after all columns used in the constraint have been defined. It is not necessary to specify them after all columns in the table are defined. Composite constraints can only be specified as table level constraints.

Various constraints that can be created on database tables are:

- **NOT NULL**

- **PRIMARY KEY**
- **CHECK**
- **UNIQUE**
- **FOREIGN KEY**

We can also specify DEFAULT value for a column. Oracle database does not consider DEFAULT as a constraint.Now let us discuss all these constraints and DEFAULT one by one in detail.

Create Table Syntax Errors

We will now look at some common errors that occur while creating tables with constraints. The statement below has several syntax errors. Let us resolve these errors step by step:

```
1 CREATE TABLE Student (
2 StudentId INTEGER PRIMARY KEY (StudentId),
3 FName VARCHAR2(10) CONSTRAINT NOT NULL,
4 LName VARCHAR2(10) CHECK (FName <> LName),
5 DOJ DATE DEFAULT,
6 Gender CHAR(1) CONSTRAINT Student_Gender_Ck CHECK Gender IN('M', 'F'),
7 PersonId INTEGER FOREIGN KEY REFERENCES Person(PersonId));
```

Display Errors

Error at line 2:

ORA-00907: missing right parenthesis

Constraints Summary

```
CREATE TABLE Student(
   StudentId INTEGER      CONSTRAINT stud_sid_pk PRIMARY KEY,C
   FName     VARCHAR2(10) CONSTRAINT stud_fname_nn NOT NULL,C3
   LName     VARCHAR2(10) NOT NULL,C1
   Gender    CHAR(1)      CONSTRAINT stud_gender_ck CHECK(Gender IN('M', 'F')),C
   DOJ       DATE DEFAULT SYSDATE,
   ContactNo NUMBER(10)   UNIQUE,C1
   PersonId  INTEGER      CONSTRAINT stud_pid_fk REFERENCES Person(PersonId),C
   CONSTRAINT stud_name_ck  CHECK(FName <> LName)T2
)
CREATE TABLE Student(
   StudentId INTEGER,
   FName     VARCHAR2(10) CONSTRAINT stud_sname_nn NOT NULL,C3
   LName     VARCHAR2(10) NOT NULL,C13
   Gender    CHAR(1),
   DOJ       DATE DEFAULT SYSDATE,
   ContactNo NUMBER(10),
   PersonId  INTEGER,
   CONSTRAINT stud_sid_pk PRIMARY KEY(StudentId),T
   CONSTRAINT stud_gender_ck CHECK(Gender IN('M', 'F')),T
   CONSTRAINT stud_name_ck CHECK(FName <> LName),T2
   UNIQUE(ContactNo),T1
   CONSTRAINT stud_pid_fk FOREIGN KEY(PersonId) REFERENCES Person(PersonId)T
)
```

C Column level constraint

T Table level constraint

1 All constraints need not be provided a name

2 Composite constraint can only be specified as table level constraint

3 Not Null can only be specified as column level constraint

Some Questions

Q1 of 6

The relationship between the two tables is created using _____

- ○ Candidate key
- ○ Primary key
- ◉ Foreign key
- ○ Check constraint

Option Foreign key is Correct

Q2 of 6

Which of the following statement(s) is/are FALSE about Primary key?

- ☐ Primary key uniquely identifies a row
- ☑ There can be more than one Primary key for a table
- ☐ Primary key column(s) in one table can be referenced by Foreign key column(s) in another table
- ☑ Columns with string(Char/Varchar2) data types cannot be made Primary key because they are large

Option There can be more than one Primary key for a table is Correct

Option Columns with string(Char/Varchar2) data types cannot be made Primary key because they are large is Correct

Q3 of 6

The entity Customer(CustId,CustName,Email,ContactNo,Adress) has three Candidate keys: a) CustId b) Email and c) ContactNo. Suggest the best Primary key for this entity.

- ◉ CustId
- ○ Email
- ○ ContactNo
- ○ CustId and ContactNo

Option CustId is Correct

Q4 of 6

RechargePlan

Operator	Plans	Amount	RechargeType
Reliance	Full-Talktime	150	Special
Airtel	Full-Talktime	140	Special
Docomo	Topup-Plan	180	Topup
Airtel	Topup-Plan	200	Topup
MTS	Topup-Plan	150	Regular
MTS	3G-Plan	300	Regular

RechargePayment

RechargeId	Operator	Plans	CardType
R1001	Reliance	Full-Talktime	Debit
R1002	Airtel	Full-Talktime	Credit
R1003	Airtel	Topup-Plan	Debit

Identify the Candidate key for the RechargePlan table.

- ○ Operator
- ○ Plans
- ○ Operator, RechargeType
- ◉ Operator, Plans

Option Operator, Plans is Correct

Q5 of 6

RechargePlan

Operator	Plans	Amount	RechargeType
Reliance	Full-Talktime	150	Special
Airtel	Full-Talktime	140	Special
Docomo	Topup-Plan	180	Topup
Airtel	Topup-Plan	200	Topup
MTS	Topup-Plan	150	Regular
MTS	3G-Plan	300	Regular

RechargePayment

RechargeId	Operator	Plans	CardType
R1001	Reliance	Full-Talktime	Debit
R1002	Airtel	Full-Talktime	Credit
R1003	Airtel	Topup-Plan	Debit

Identify the Primary key for the RechargePayment table.

- ○ Operator
- ○ Plans
- ◉ RechargeId
- ○ Operator, Plans

Option RechargeId is Correct

Q6 of 6

RechargePlan

Operator	Plans	Amount	RechargeType
Reliance	Full-Talktime	150	Special
Airtel	Full-Talktime	140	Special
Docomo	Topup-Plan	180	Topup
Airtel	Topup-Plan	200	Topup
MTS	Topup-Plan	150	Regular
MTS	3G-Plan	300	Regular

RechargePayment

RechargeId	Operator	Plans	CardType
R1001	Reliance	Full-Talktime	Debit
R1002	Airtel	Full-Talktime	Credit
R1003	Airtel	Topup-Plan	Debit

Identify the Foreign key for the table RechargePayment.

- ○ Operator
- ○ Plans
- ○ RechargeId, Operator
- ◉ Operator, Plans

Option Operator, Plans is Correct

Q1 of 3

In a relationship between two entities, one and only one instance of the first entity is related to zero or many instances of the second entity. Which of the following symbols in Crows Feet notation represents this relationship?

- ○ A
- ◉ B
- ○ C
- ○ D

Option B is Correct

Q2 of 3

In a many to one relationship, the primary key of one entity acts as foreign key on which side?

- ○ On the side where single (one) relationship is defined
- ◉ On the side where many relationship is defined
- ○ On both the sides
- ○ Neither of them

Option On the side where many relationship is defined is Correct

Q3 of 3

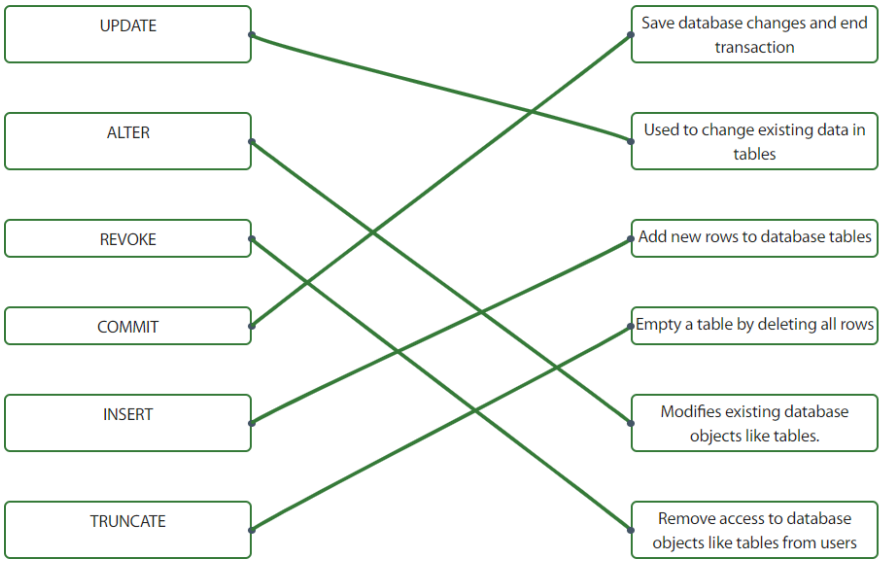
A many to many relationship between two entities usually results in how many tables?

- ○ Two
- ◉ Three
- ○ Four
- ○ Five

Option Three is Correct

Q8 of 8

Match the following commands and the descriptions.



Q1 of 9

Choose a data type to store weight of a person in kilograms with accuracy of 1 gram. As per wikipedia, Jon Brower Minnoch was the heaviest human ever recorded with a weight of 635kg.

- ○ NUMBER(3,3)
- ○ NUMBER(3,2)
- ○ NUMBER(6,2)
- ◉ NUMBER(6,3)

Option NUMBER(6,3) is Correct

Q2 of 9

What is the maximum value that can be stored in NUMERIC(4,2)?

- ○ 9999.99
- ○ 99.9999
- ◉ 99.99
- ○ 9.99

Option 99.99 is Correct

Q3 of 9

Which of the following statements are TRUE?

- ☑ Scale denotes number of digits allowed after decimal point
- ☐ Precision denotes number of digits allowed before decimal point
- ☑ SQL performs rounding if user attempts to store a value that has higher scale than the data type
- ☐ SQL performs truncation if user attempts to store a value that has higher precision than the data type

Option Scale denotes number of digits allowed after decimal point is Correct

Option SQL performs rounding if user attempts to store a value that has higher scale than the data type is Correct

Q4 of 9

Choose the most suitable data type possible for the given column.

Column Name: Price

Description: Cost of an item in rupees and paise

Example: 200.21

- ○ VARCHAR2(50)
- ○ NUMBER
- ◉ NUMBER(5,2)
- ○ NUMBER(6)

Option NUMBER(5,2) is Correct

Q5 of 9

Choose the most suitable data type possible for the given column.

Column Name: Meeting_Time

Description: Time and date of the meeting

Example: 2014/01/01 10:00 AM

- ○ DATE
- ◉ TIMESTAMP
- ○ BLOB

Q6 of 9

Choose the most suitable data type possible for the given column.

Column Name: Profile_Image

Description: Image of the employee

- ○ CLOB
- ◉ BLOB

Option BLOB is Correct

Q7 of 9

Choose the most suitable data type possible for the given column.

Column Name: Book_Title

Description: A text string

Example: Winning

- ◉ VARCHAR2(50)
- ○ CHAR(11)
- ○ INTEGER

Option VARCHAR2(50) is Correct

Q8 of 9

Choose the most suitable data type possible for the given column.

Column Name: Gender

Description: A single character gender code, M or F

Example: M

- ○ CHAR(11)
- ○ VARCHAR2(50)
- ◉ CHAR(1)

Option CHAR(1) is Correct

Q9 of 9

Choose the most suitable data type possible for the given column.

Column Name: PIN_Code

Description: Six digit numeric PIN code for any address in India

Example: 560100

- ○ INTEGER
- ○ CHAR(11)
- ◉ NUMBER(6)
- ○ VARCHAR2(50)

Option NUMBER(6) is Correct

Q1 of 1

Which of the following statement is TRUE regarding CREATE TABLE statement?

- ○ CREATE table replaces the existing table if it already exists
- ○ Attributes default to NUMBER data type if data type is not provided
- ◉ Attributes allow NULL values unless NOT NULL clause is provided
- ○ DEFAULT clause can only be provided for NOT NULL attributes

Option Attributes allow NULL values unless NOT NULL clause is provided is Correct

Q1 of 4

Which requirements can be implemented using a referential integrity constraint?

- ○ Customer must have a name
- ○ Customer must be greater than 18 years old
- ◉ Customer information must be known before anything is sold to him/her
- ○ Two customers cannot have same mobile number

Option Customer information must be known before anything is sold to him/her is Correct

---

Q2 of 4

Which of the following statements are TRUE?

- ☐ A column with a CHECK constraint does not allow NULL values
- ☑ A Unique constraint allows multiple rows to have NULL value
- ☐ A PRIMARY KEY allows a single row to contain NULL
- ☑ Referential integrity constraint allows NULL value

Option A Unique constraint allows multiple rows to have NULL value is Correct

Option Referential integrity constraint allows NULL value is Correct

Q3 of 4

Which requirements can be implemented using a CHECK constraint?

- ☐ Customer must have a name
- ☑ Customer must be greater than 21 years old
- ☑ Customer must have a residence in an Asian Country
- ☐ Two customers cannot have same email id

Option Customer must be greater than 21 years old is Correct

Option Customer must have a residence in an Asian Country is Correct

---

Q4 of 4

Which constraint can be defined only at the column level?

- ○ UNIQUE
- ◉ NOT NULL
- ○ DEFAULT
- ○ PRIMARY KEY

Option NOT NULL is Correct

Q1 of 3

What will be the output of the following command?

```
CREATE TABLE Emp (
    Name VARCHAR2(20) PRIMARY KEY,
    DOB DATE PRIMARY KEY,
    DeptNo NUMBER(4));
```

- ◯ Two primary keys will be created
- ◯ Composite primary key will be created with NAME as first attribute
- ◉ Error will be thrown
- ◯ Composite primary key will be created with DOB as first attribute

Option Error will be thrown is Correct

Q2 of 3

Consider the creation of the following table: Customer(CustId, AccountNo, CustName) with combination of columns Custid and AccountNo should be UNIQUE. Which one of the following options is CORRECT?

- ◯ Put UNIQUE constraint on both the columns separately
- ◯ This type of constraint is not possible
- ◯ Put PRIMARY KEY constraint on both the columns separately
- ◉ Put a table level UNIQUE constraint involving both the columns

Option Put a table level UNIQUE constraint involving both the columns is Correct

Q3 of 3

Which one of the following DDL statements will create a table? Assume table customer is already created with appropriate constraints and data types.

a.

```
CREATE TABLE Account(
    Acctid NUMBER(3),
    Custid NUMBER(4),
    Balance NUMBER(8,2),
    CONSTRAINT ac_aid_pk PRIMARY KEY(Acctid),
    CONSTRAINT ac_cid_fk FOREIGN KEY(Custid) REFERENCES Cust
```

b.

```
CREATE TABLE Account(
    Acctid NUMBER(3) PRIMARY KEY(Acctid),
    Custid NUMBER(4) REFERENCES Customer(Cid),
    Balance NUMBER(8,2));
```

c.

```
CREATE TABLE Account(
    Acctid NUMBER(3) CONSTRAINT PRIMARY KEY,
    Custid NUMBER(4) CONSTRAINT ac_cid_fk FOREIGN KEY(Custid
    Balance NUMBER(8,2) );
```

- (●) a
- ( ) b
- ( ) c

Option a is Correct