Database Management System by scaler   
----------DBMS----------------  
er model and diagrams  
introduction to relational models  
relational algebra  
relational calculus  
sql intro  
sql commands  
sql joins  
sql advanced queries  
sql dml,ddl,dcl,  
functional dependencies and decomposition  
normalization  
transaction and concurrency control  
file structure(b and b+)  
(pls recall this regularly before new session)  
nov 2:  
 Database management system is a software which is used to interact with the database. It also creates, deletes and manages file in the form of table when compare to file system it is easy to access data.   
file system requires large number of lines of code to retrieve data where as dbms uses sql query to retrieve data which is simple to follow.  
for retrieving data from the file it need to traverse entire file one by one whereas in dbms it follows powerful method called 1) indexing, which retrieves data from dbms in faster and easy way with sql queries.

2) Redundancy: repeatedly storing same data  
for example: we have customer table(cId, cName, cAdd,…) and purchase table(cId, pId, pName…)  
we no need to store entire data of customer details once again in purchase table we can make link of cId from both the tables.  
This is also possible in file system but it requires more code.  
  
3) Consistency:  
 table designing matters whenever new purchase happens it automatically update the system.  
  
4) Data independence: It hides low level data from the engineer using dbms like where, how it stores in what way it retrieves data  
  
5)Security and access control and Abstraction: Hides all data structure and algorithm things sql is enough  
  
nov-3  
tables,keys,schema: It all deals with relational database  
 columns/fields/attributes and rows/tuple/record  
whole table is called as instance  
Instance is called as set of tuple,rows along with the table structure  
Key: minimum set of attribute/ columns to uniquely identify a row/tuple  
**Simple key:** key with only one attribute (cId)  
**Compound key:** key with multiple attributes(cName,cAddress)  
**Candidate key:** collection of all unique key {cId,( cName,cAddress)}  
candidate key can be null  
**Primary Key:** one of the candidate key that db-designer chooses to maintain uniqueness  
 it should be not null, primary key need to be only one in a table  
 it should be unique in table these are the bunch of constrains in a primary key  
the above mentioned constrains are named as entity integrity constrains  
Alternate/Secondary key: It is candidate key but it is not unique(ie primary key)

Super Key: candidate key (union) attributes  
(cId, cName) = (cId) u (cName)  
foreign key: it refers the primary key of other table. The value which found in child table need to be available in parent table too. Otherwise it violates Integrity Constrain  
Self referential key: it refers the column in its own table

**Types of Integrity Constrain:**1. Entity Integrity, 2. Domain Integrity 3. Referential Integrity 4. User defined Integrity  
1. It follows the rules of primary key contrains like unique and not null, table need to contain only one primary key  
2. Db admin need to design how column or attribute need to follow the set of valid values clearly for the particular Range  
4. It deals with referenced table and referencing table.  
if we insert any value in rfd table no changes require  
if we delete it follows 3 options: on delete no action (it does not require any changes in rfng table)  
 on delete cascade (it parallely deletes corresponding column in rfng table it deletes as chain of values in it)  
 on delete not null( it deletes and set null value in rfng table)  
3. In purchase table amazon will not allow customer to purchase a same product after certain limit it sets condition or constrain in it.

Nov-4  
Entity Relationship Model:  
 high level picture of the database -table,key,Integrity constrains  
Entity: Object that can be uniquely identified  
**Attribute** – ppt/ feature of an entity/ relationship  
**key attribute**: primary key of a table (represents circle underline)  
**multi attribute** - customer have multiple phone no,payment method ( double circle)  
**derived attribute** – age of the person can be derived from dob ( dotted circle)  
**compound attribute:** It have all pieces like customer name may have fname,lname,mname

**Relationship**: represents diamond. It is used to connect multiple entities.   
Total and partial Participation.

Nov-5  
Cardinality of Relationship: One to one, one to many, many to one, many to many  
one to one: one customer will have only one driving license( it denotes arrow in both sides with has a relationship)  
one to many: one manager can manage many teams (if it denotes single line it is partial, double line total participation)  
many to one: many credits can be owned by one customer  
many to many: many customer can buy many product

--------------------------------------- **Database Management System** ---------------------------------------

**Database**  
 It is organized collection of data that can be modified,retrieved, or updated  
 The data stored in db can be column and row format as a table, it is basically a box which will store related items

**Types of DB:**  
object oriented db: It represents data in the form of class and object. Object -real world entity, class – collection of objects eg: Postgresql  
relational db: collection of data item with predefined relation between them  
distributed db: It is an integrated collection of db, that is physically distributed across sites in a computer network. Eg: Cassandra, mongodb  
hiererachical db: data stored in the form of records which forms tree like structures  
 eg: IBM -Information Management System  
**DBMS:** It is a software which acts as a interface between data and application and also used to store and retrieve data and also it allows accessing the data from the database.  
Characteristics: real world entity, rdbms, sql, isolation of data and app, acid properties, multiuser and concurrent access

**Single tier Architecture:** It is simple architecture which resides everything in single machine like client server and database. It is used to practice sql queries anytime. Once the machine is installed it can setup by using commands in the prompt.

**2 tier Architecture:** Presentation Layer runs on client (mobile, pc, laptop) and data is stored in server called second tier. This architecture provides security to the dbms since it is not exposed to end user directly.   
 It provides fast and direct communication.

**3 tier Architecture:** It is most popular client server architecture in which development, maintenance of functionality process, logic, access to storage of data, user interface all these are performed in separate layers ie presentation, application and database server.

**Schema:** It is the blueprint of a table and description of a table  
Physical Schema: It deals with actual storage of data like files, indices and defines how data is stored in secondary storage.  
logical database: It has logical constrains that need to be applied all the time of data stored and also it defines view, indecies and also includes integrity constrains.  
  
ER Model: It is represention of table which is db with relationship between the tables.  
 Entity – noun, place or object

Weak Entity: it need to depend on other entity. Eg installment needs to depend on loan.

Attribute: ppt of an entity, key attribute: primary key (student id)  
composite attribute: which is composed of many other attributes(name like: fname, mname, lname)  
multiple attribute: which is composed of more than one attribute( single person have many phno)  
Derived Attribute: It is derived from existing column  
  
**Relational database:** It is type of db which stores and provides access to datapoints that stores and provide access to datapoints that are related to each other.  
Rdb each row inn a table is a record with unique id called key.  
It is structured, easy to access data.

**Transaction:** It is the single logical unit of work which accesses and possibly modifies the content in the database.  
**Transaction state**  
Active state: If the transaction is in execution, it will remain active no matter the execution is in which step of progress, it will remain active.  
Failure state: If a transaction is in execution when failure occurs, it goes from active state to fail state.

Acid properties:   
Atomicity: Either the transaction takes place once or does not happen at all then there is no midway ie it will occur partially  
 It consider each transaction as one unit either it will be completed or not at all executed.  
 It has two operations: commit and abort  
Consistency: In short update, once the transaction is done it need to update the db as before and after  
Isolation: At the same time multiple users can use or access same db, modify it will perform concurrently.  
Durability: This ppt ensures once the transaction is done it will store the modification in memory and it will retain even if any failure occur it will permanently stores in non volatile memory. The update becomes permanent

Concurrency: In multiprogramming environment where multiple transaction can be executed simultaneously, it is highly important to control the concurrency of transaction.

we have concurrency control protocol to ensure atomicity, isolation and serialzability of concurrent transactions

Normalization: It is technique which is used to avoid duplication and data redundancy  
 efficient data access, find info quickly, more compact database.  
1nf: each table cell should have single value, all record must be unique.  
2nf: db should be 1 nf and should also have a single primary key.  
3nf: db should be in 2nf and must not have any transitive independencies.