Last Minute Notes – DBMS

Difficulty Level : Medium   ●   Last Updated : 28 Jun, 2021

See Last Minute Notes on all subjects here.

We will discuss the important key points useful for GATE exams in summarized form. For details you may refer this.

E-R Diagram: The most common asked questions in ER diagram is minimum number of tables required for a given ER diagram. Generally, following criteria are used:

Cardinality	Minimum No. of tables
1:1 cardinality with partial participation of both entities	2
1:1 cardinality with total participation of atleast 1 entity	1

Related Articles   **Save for later**

Note: This is a general observation. Special cases need to be taken care. We may need extra table if attribute of a relationship can't be moved to any entity side.

Keys of a relation: There are various types of keys in a relation which are:

- **Candidate Key:** The minimal set of attributes which can determine a tuple uniquely. There can be more than 1 candidate key of a relation and its proper subset can't determine tuple uniquely and it can't be NULL.
- **Super Key:** The set of attributes which can determine a tuple uniquely. A candidate key is always a super key but vice versa is not true.
- **Primary Key and Alternate Key:** Among various candidate keys, one key is taken primary key and others are alternate keys.
- **Foreign Key:** Foreign Key is a set of attributes in a table which is used to refer the primary key or alternative key of the same or other table.

Normal Forms

- **First Normal Form:** A relation is in first normal form if it does not contain any multi-valued or composite attribute.
- **Second Normal Form:** A relation is in second normal form if it does not contain any partial dependency. A dependency is called partial dependency if any proper subset of candidate key determines non-prime (which are not part of candidate key) attribute.
- **Third Normal Form:** A relation is in third normal form if it does not contain any transitive dependency. For a relation to be in Third Normal Form, either LHS of FD should be super key or RHS should be prime attribute.
- **Boyce-Codd Normal Form:** A relation is in Boyce-Codd Normal Form if LHS of every FD is super key. The relationship between Normal Forms can be represented as: **1NF⊃2NF ⊃3NF ⊃BCNF**

Relational Algebra: Procedural language with basic and extended operators.

Basic Operator	Semantic
σ(Selection)	Select rows based on given condition
∏(Projection)	Project some columns
X (Cross Product)	Cross product of relations, returns **m*n** rows where m and n are number of rows in R1 and R2 respectively.

▲

U (Union)	Return those tuples which are either in R1 or in R2. Max no. of rows returned = **m+n** andMin no. of rows returned = **max(m,n)**
−(Minus)	R1-R2 returns those tuples which are in R1 but not in R2. Max no. of rows returned = **m** and Min no. of rows returned = **m-n**
ρ(Rename)	Renaming a relation to other relation.

◀ ▲ ▶

| **Extended Operator**	**Semantic**
∩ **(Intersection)**	Returns those tuples which are in both R1 and R2. Max no. of rows returned = min(m,n) and Min no. of rows returned = 0
⋈$_c$**(Conditional Join)**	Selection from two or more tables based on some condition (Cross product followed by selection)
⋈**(Equi Join)**	It is a special case of conditional join when only equality condition is applied between attributes.
⋈**(Natural Join)**	In natural join, equality condition on common attributes hold and duplicate attributes are removed by default. **Note:** Natural Join is equivalent to cross product if two relations have no attribute in common and natural join of a relation R with itself will return R only.
⋈**(Left Outer Join)**	When applying join on two relations R and S, some tuples of R or S does not appear in result set which does not satisfy the join conditions. But Left Outer Joins gives all tuples of R in the result set. The tuples of R which do not satisfy join condition will have values as NULL for attributes of S.
⋈**(Right Outer Join)**	When applying join on two relations R and S, some tuples of R or S does not appear in result set which does not satisfy the join conditions. But Right Outer Joins gives all tuples of S in the result set. The tuples of S which do not satisfy join condition will have values as NULL for attributes of R.

▲

⋈(Full Outer Join)	When applying join on two relations R and S, some tuples of R or S does not appear in result set which does not satisfy the join conditions. But Full Outer Joins gives all tuples of S and all tuples of R in the result set. The tuples of S which do not satisfy join condition will have values as NULL for attributes of R and vice versa.
/(Division Operator)	Division operator A/B will return those tuples in A which is associated with every tuple of B.**Note:**Attributes of B should be proper subset of attributes of A. The attributes in A/B will be Attributes of A- Attribute of B.

◄   ►

How to solve Relational Algebra problems for GATE – SET 1
How to solve Relational Algebra problems for GATE – SET 2

SQL: As opposed to Relational Algebra, SQL is a non-procedural language.

Operator	Meaning
Select	Selects columns from a relation or set of relations.**Note:** As opposed to Relational Algebra, it may give duplicate tuples for repeated value of an attribute.
From	**From** is used to give input as relation or set of relations from which data needs to be selected.
where	**Where** is used to give condition to be used to filter tuples
EXISTS	**EXISTS** is used to check whether the result of a correlated nested query is empty (contains no tuples) or not.
Group By	**Group By** is used to group the tuples based on some attribute or set of attributes like counting the no. of students group by department.
Order By	**Order By** is used to sort the fetched data in either ascending or descending according to one or more columns.
Aggregate functions	Find the aggregated value of an attribute. Used mostly with group by. e.g.; count, sum, min max. **select count(*) from student group by dept_idNote:** we can select only those columns which are part of group by.
Nested Queries	When one query is a part of other query. Solving nested queries questions can be learnt in

▲

https://www.geeksforgeeks.org/nested-queries-in-sql/

◀   ▶
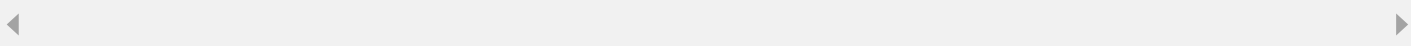
Conflict serializable and Conflict Equivalent: A schedule is conflict serializable if it is conflict equivalent to a serial schedule.

Checking for Conflict Serializability

To check whether a schedule is conflict serializable or not, find all **conflicting operations pair**s of a schedule and draw precedence graph (For all conflicting operation pair, an edge from T_i to T_j if one operation of conflicting pair is from T_i and other from T_j and operation of T_i occurs before T_j in schedule). If graph does not contain cycle, the schedule is conflict serializable else it is not conflict serializable.

Schedules are said to be conflict equivalent if 1 schedule can be converted into another by swapping non conflicting operations.

Note: Two phase locking protocol produce conflict serializable schedule but may suffer from deadlock. On the other hand, Time-Stamp based protocols are free from deadlock yet produce conflict serializable schedule.

View Serializable and View Equivalence : Two schedules S1 and S2 are said to be view-equivalent if all conditions are satisfied for all objects:

- If the transaction T_i in S1 reads an initial value for object X, in S2 also, T_i must read the initial value of X.

- If the transaction T_i in S1 reads the value written by transaction T_j in S1 for object X, same should be done in S2.

- If the transaction T_i in S1 is the final transaction to write the value for an object X, in S2 also, T_i must write the final value of X.

A schedule is view serializable if it is view equivalent to any serial schedule.

Irrecoverable Schedules: For a transaction pair $< T_i, T_j >$, if T_j is reading the value updated by Ti and Tj is committed before commit of Ti, the schedule will be irrecoverable.

▲

Recoverable Schedules: For a transaction pair $< T_i, T_j >$, if T_j is reading the value updated by Ti and Tj is committed after commit of Ti, the schedule will be recoverable.

Cascadeless Recoverable Schedules: For a transaction pair $< T_i, T_j >$, if value updated by Ti is read by Tj only after commit of T_i, the schedule will be cascadeless recoverable.

Strict Recoverable: For a transaction pair $< T_i, T_j >$, if value updated by Ti is read or written by Tj only after commit of T_i, the schedule will be strict recoverable. The relationship between them can be represented as:

Strict $\subset$ **Cascadeless Recoverable** $\subset$ **recoverable** $\subset$ **all schedules**

File structures

Primary Index :: A primary index is an ordered file, records of fixed length with two fields. First field is same as primary key as data file and second field is a pointer to data block, where the key is available.

The average number of block accesses using index = $\log_2 \mathbf{Bi + 1}$, where Bi = number of index blocks.

Clustering Index : Clustering index is created on data file whose records are physically ordered on a non-key field (called Clustering field).

Secondary Index : Secondary index provides secondary means of accessing a file for which primary access already exists.

```
Number of index entries = Number of records
```

B Trees

At every level , we have Key and Data Pointer and data pointer points to either block or record.

Properties of B-Trees :

Root of B-tree can have children between **2** and **P**, where P is Order of tree.

Order of tree – Maximum number of children a node can have.

Internal node can have children between [**P/2**] and **P**
Internal node can have keys between [**P/2**] **– 1** and **P-1**

B+ Trees

In B+ trees structure of leaf and non-leaf are different, so their order is. Order of non-leaf will be higher as

compared to leaf nodes.

Searching time will be less in B+ tress, since it doesn't have record pointers in non-leaf because of which depth will decrease.

This article has been contributed by Sonal Tuteja.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Attention reader! Don't stop learning now. Get hold of all the important CS Theory concepts for SDE interviews with the **CS Theory Course** at a student-friendly price and become industry ready.

Like

Previous   Next

Commonly asked DBMS interview   **How to make a website using WordPress**
questions | Set 2   **(Part – 2)**

▲