

1. a) 19-20(1-a)
 b) What is the difference between a Candidate key, a primary key & Composite key? What considerations might influence the choice of a primary key?

Primary	Candidate	Composite
i. There can be only one primary key in any table/relation ii. The primary key is unique & not null. iii. Only one Candidate key can be primary key iv. Ex: Student(ID, Name, address) PK - ID (✓ To denote a CK ✓ Chosen with care by DBA ✓ Never or very rarely changed ✓ CK with NULL value is ✓ NOT the primary key ✓ PK =	i. There can be multiple / more than one Candidate key in a table/relation ii. Whereas a Candidate unique & alternate keys can be null. iii. Each CK can qualify as PK iv. Ex: Stu(ID, Name, email) CK: ID, email (minimal super keys are)	i. Two or more attributes (columns) that together can uniquely identify a tuple (record) in a table and work as PK. ii. Combination of columns iii. Creating more than one primary key is jointly known as Composite key.

b) 10-20

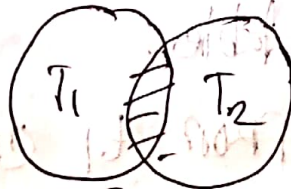
c) What are the Join type and Condition that are permitted in SQL?

A join clause is used to combine rows from two or more tables, based on a related column between them.

Here are different type of join in SQL:

i. INNER JOIN: Return records that have matching values in both tables.

Syntax: `SELECT C-name(s) FROM t1 INNER JOIN t2 ON t1.C-name = t2.C-name`



Inner

ii. Left (Outer): Returns all records from the left table and the matched records from the right table.



Syntax: `SELECT * FROM t1 LEFT JOIN t2 ON "`

- ✓ Right (Outer): Returns all records from the right table and the matched records from the left table



Syntax: `SELECT * FROM t1 RIGHT JOIN t2 ON` —

- ✓ Full (Outer): Return all records when there is a match in either left or right table.



Syntax: `SELECT * FROM t1 FULL JOIN t2 ON` ;

- ✓ CROSS join: Returns the Cartesian Product of two tables, i.e. all possible combination of rows from both tables.

Syntax: `SELECT * FROM t1 CROSS JOIN t2;`

- ✓ SELF JOIN: Join a table with itself, often used when working with hierarchical data or to compare rows within the same table.

Syntax: `SELECT * FROM t1 INNER JOIN t2 ON t1.Column = t2.Column;`

Natural join: Performs a join based on columns with the same name in both tables.

Syn: `SELECT * FROM t1
NATURAL JOIN t2;`

a) diff between Specialization & generalization.

Specialization

Generalization

i. This process operates in a top-down manner

i. This process operates in a bottom manner

ii. The entity of higher-level must always have the lower level entity (generalization)

ii. The entities of the higher level might not consist of the entities of the lower level. (Specialization)

iii. The Specialization process takes up an entity and splits to form various new entities each of which inherits some of the features of the splitting entity.

iii. The gen process extracts the most common features of various entities and leads to the formation of a new entity.

Specialization

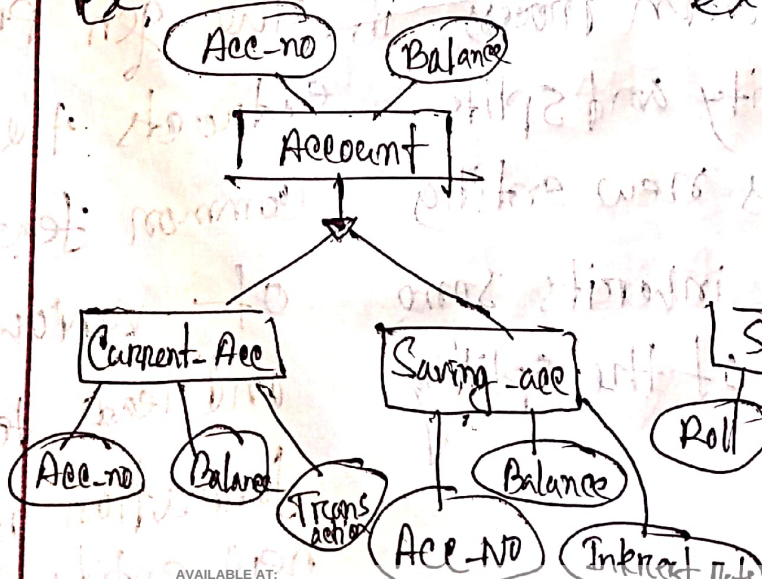
iv. A user can easily apply a generalization to a group of different entities

v. It leads to a reduction in the size of a schema

vi. This process leads to the formation of a single entity out of multiple entities

vii. There is inheritance in Spec

ex:



Generalization

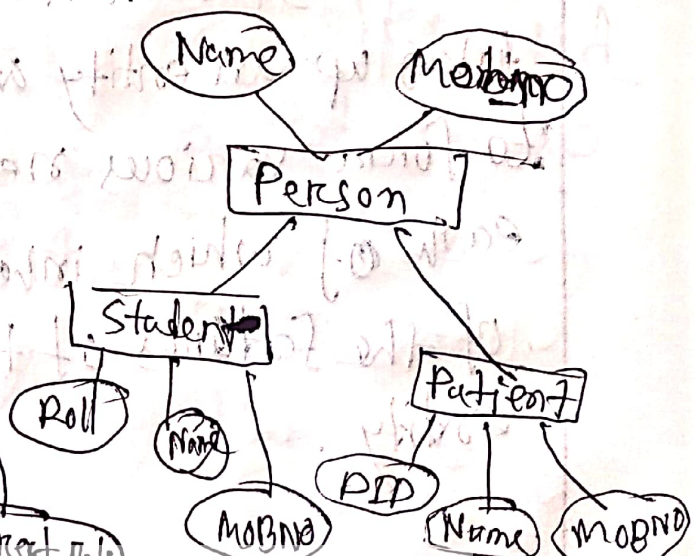
iv. we can only apply specialization to a single entity.

v. It leads to an increase in the size of a schema

vi. This process leads to the formation of multiple entities from one single entity.

vii. There is no inheritance in Gene-

ex:



b) Same as 2019-20 (2-b)

c) Same as 2018-19 (2-b)

d) Draw the ER diagram for ...

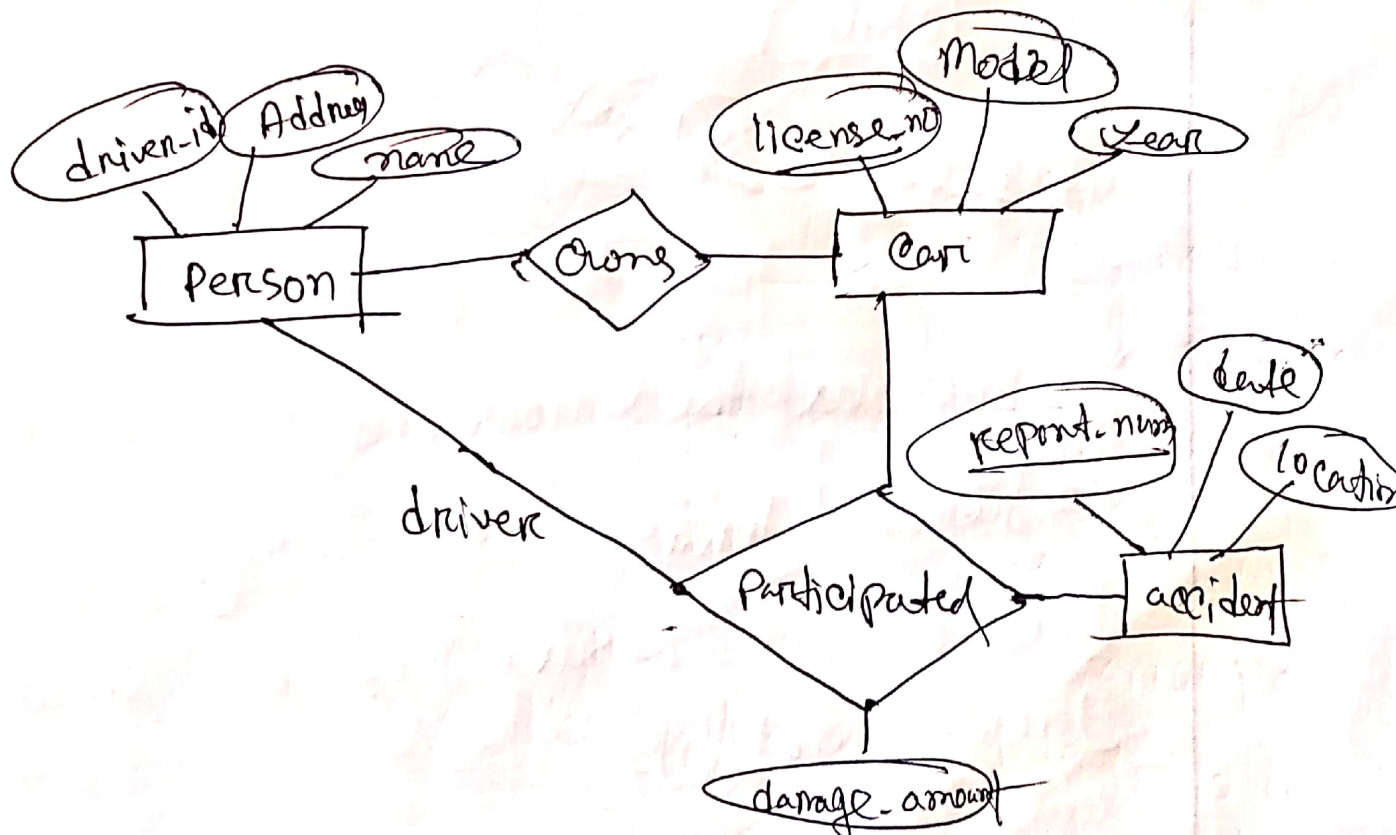
Person (driver-id, name, address)

Car (license no, year, model)

Accident (report number, date, location)

owns (driver-id, license-no)

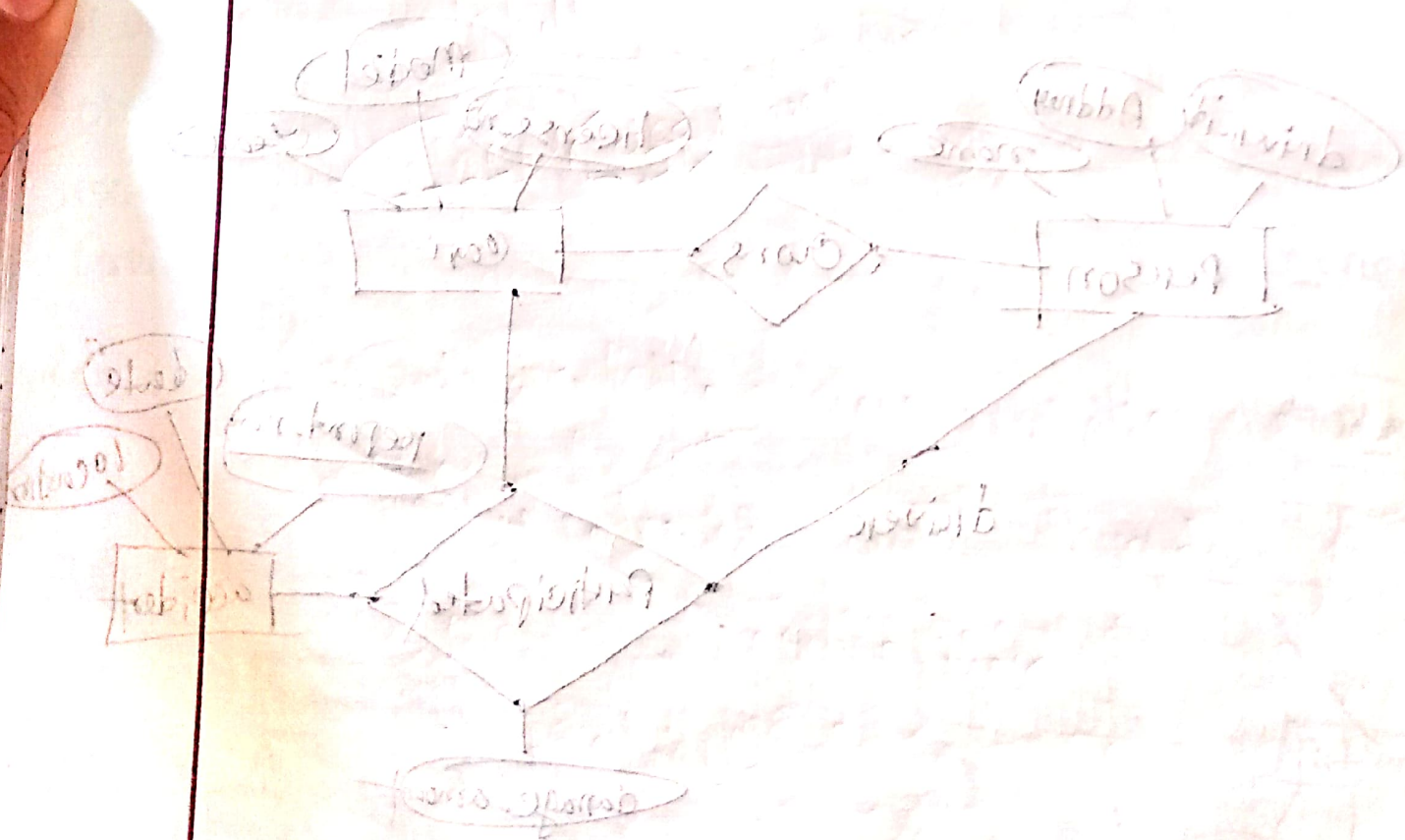
Participated (driverid, license-no, report num,
damage amount)



3. a) $10 - 20 (2-a)$

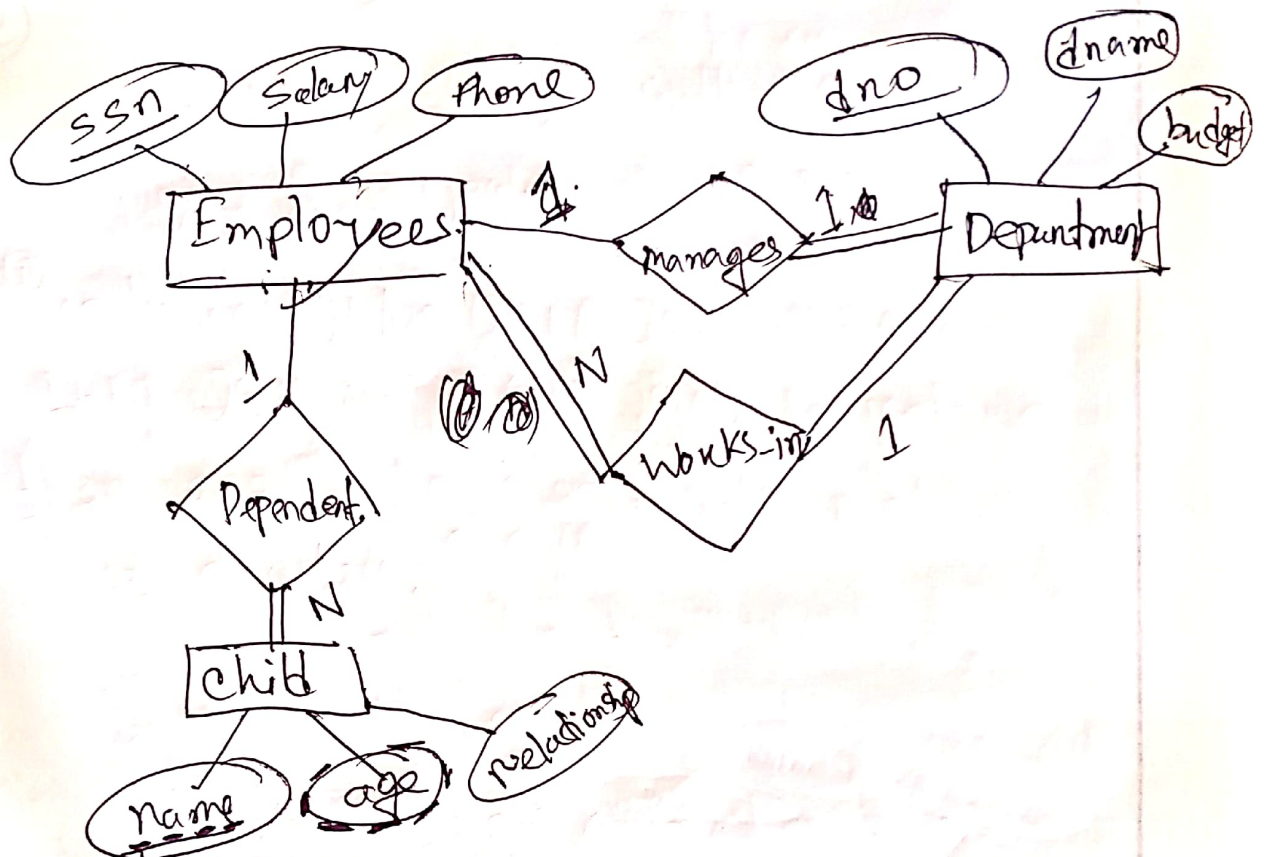
b) $10 - 20 (2-d)$

c)



4. a) 19-20 (4-a)

b)



6. a) What are the various anomalies we may have to face in retrieving data through joining relations? How we can resolve those through various joining operations?

Anomaly means inconsistency in the pattern from the normal form. When retrieving data through joining relations in a relational database, various anomalies may arise.

i. Update anomalies;

☐ Definition:- Update anomalies occur when updating data in one relation leads to inconsistencies or unintended changes in other related relation.

☐ Ex - Let you have a database of employees & projects. If an employee changes their department

☐ Resolution - Use INNER JOIN or OUTER JOIN operations based on the specific requirement of your query.

ii. Insert Anomalies

☐ Definition - Insert anomalies occur when adding new data to a table is not possible without adding data to another related table, leading to incomplete or inaccurate information.

5. a) ~~1000~~ SQL PDF

b) 2019-20 (5-b)

6. a) Def - In a database with separate tables for employees & projects.

Resolution: Utilize OUTER JOIN operation. Such as LEFT JOIN or RIGHT JOIN, to retrieve data from one table along with matching data from another.

iii. Delete Anomalies:

Def: Delete anomalies occur when deleting data from one relation unintentionally removes related data in other tables, leading to the loss of information.

Ex: If you delete an employee record and that employee is associated with only one project in another table,

Resolution: Use INNER and OUTER JOIN operations - to avoid unnecessary data loss during deletion.

5) Ans:)
The given relational schema for the "Department" table has a problem that is each department can have multiple locations, which violates the 1st 1NF requirement. In 1NF each attribute in a relation must be atomic (can't have multiple values).

Department:

dept number	dept name	dept manager no	dept location
101	CSF	001	Barishal
102	MATH	002	Dhaka
103	ENG	003	Rajshahi, Dhaka

To achieve 1NF for such a relation, need to restructure the schema by creating a separate table for dept. locations.

Department:

dept number	dept name	dept manager no
101	CSF	001
102	MATH	002
103	ENG	003

AVAILABLE AT:

Onebyzero Edu - Organized Learning, Smooth Career

The Comprehensive Academic Study Platform for University Students in Bangladesh (www.onebyzeroedu.com)

Department location:

dept number	dept location
101	Banishel
101	Khulna
102	Dhaka
102	Khulna
102	Rajshahi
103	Khulna

In this modified scheme,

- i. The "Department" table now contains only attributes that are atomic
- ii. The "Dept-location" table is introduced to handle the multiple locations associated with each dept. Composite pk — dept number & dept location. For this table

②) FDS hold in the relation R₉

i. $AB \rightarrow C$

ii. $B \rightarrow D$

iii. $DE \rightarrow A$

A	B	C	D	E
1	2	3	4	5
1	4	3	4	5
1	2	4	4	1

Consider $R(A, B)$ with the following instance of R on this instance $\alpha \rightarrow \infty$ holds
 $\infty \rightarrow \beta$ does not hold.

① $AB \rightarrow C$

Suppose $AB = \alpha$ & $C = \beta$

$t_1[\alpha] = 1$ $t_2[\alpha] = 1$ $t_3[\alpha] = 1$

$t_1[\beta] = 3$ $t_2[\beta] = 3$ $t_3[\beta] = 4$

for tuple 3 $\beta \rightarrow \alpha$ does not hold

so, $AB \rightarrow C$ is not possible

② $B \rightarrow D$

$t_1[\alpha] = 2$ $t_2[\alpha] = 4$ $t_3[\alpha] = 2$

$t_1[\beta] = 4$ $t_2[\beta] = 4$ $t_3[\beta] = 4$

$B \rightarrow \alpha$ is hold but $\alpha \rightarrow \beta$ does not hold

so, $B \rightarrow D$ is possible on holds in R

(19)

$DE \rightarrow A$

$$t_1[A] = 9, 5$$

$$t_2[A] = 9, 5$$

$$t_3[A] = 9, 1$$

$$t_1[B] = 1$$

$$t_2[B] = 1$$

$$t_3[B] = 1$$

So, $DE \rightarrow A$ holds in R .

d)

FDS: $\{AB \rightarrow CD, BC \rightarrow D\}$. Find minimal cover on irreducible set.

Soln:

Break down the following FDs:

$$* AB \rightarrow C$$

$$AB \rightarrow D$$

$$BC \rightarrow D$$

Using these dependencies:

$$i. AB^+ = \{A, B, C, D\}$$

$$AB^+ = \{A, B, D\} \text{ [without } AB \rightarrow C]$$

So, the closure is changing

So,

$$AB \rightarrow A$$

$$AB \rightarrow B$$

$$AB \rightarrow D$$

$$AB \rightarrow C$$

$$AB \rightarrow A$$

$$AB \rightarrow B$$

$$AB \rightarrow D$$

$$AB \rightarrow C$$

$$AB^+ = \{A, B, D\}$$

$$AB^+ = \{A, B, D\} \text{ [without } AB \rightarrow D]$$

So, $AB \rightarrow D$ is essential

$$\text{in, } BC^+ = \{B, C, D\}$$

$$BC^+ = \{B, C\} \text{ [without } BC \rightarrow D]$$

So, $BC \rightarrow D$ is essential

So, Canonical Coven

$$AB \rightarrow C$$

$$A \rightarrow B$$

$$BC \rightarrow D$$

Union

~~Transitivity rule~~

$$AB \rightarrow D$$

7

a) Why should NULL in a relation be avoided as possible? Explain

Ans. Avoiding the use of NULL values in a relation is a good practice in DB design for several reasons:

i. Semantic Ambiguity: Null values can introduce semantic ambiguity, as they may represent different things in different contexts. This lack of clarity can lead to misunderstanding and errors in data interpretation.

ii. Complexity in Querying: Handling NULL values in queries adds additional logic, such as using the 'IS NULL' or 'IS NOT NULL' conditions. This complexity makes queries more difficult to read, write & maintain.

iii. Aggregation Challenges: NULL values can cause challenges when aggregating data. Operators like SUM, AVG, COUNT etc. may not provide accurate results when NULL values are involved.

iv. Joining & Comparison Issues: Null values can cause challenges in joining and comparing data.

operations & Comparisons.

vi. Indexing & Searching: Indexing on columns with NULL values. ~~It~~ may not be as efficient as indexing on columns with non-NULL values.

vii. Data Integrity: Null values can make it challenging to enforce data integrity constraints.

viii. Compatibility Issues: Different DB system may handle NULL values differently.

ix. Three-Valued logic: SQL uses a three-valued logic system that includes True, False & Unknown (representing NULL).

b) i) let's analyze the given transactions:

$T_1: R(A); R(B); \text{if } A=0 \text{ then } B := B+1; W(B)$
 $T_2: R(B); R(A); \text{if } B=0, \text{ then } A := A+1; W(A);$

The consistency requirements is $A=0 \vee B=0$ with the initial values $A=0$ & $B=0$.

7. a) There are two possible executions:
 $T_1 T_2$ & $T_2 T_1$

Case 1:

	A	B
initially	0	6
after T_1	0	11
after T_2	0	1

Consistency met: $A=0 \vee B=0 \Rightarrow T \vee F \Rightarrow T$

Case 2:

	A	B
initially	0	0
after T_2	1	0
after T_1	1	0

Consistency met:

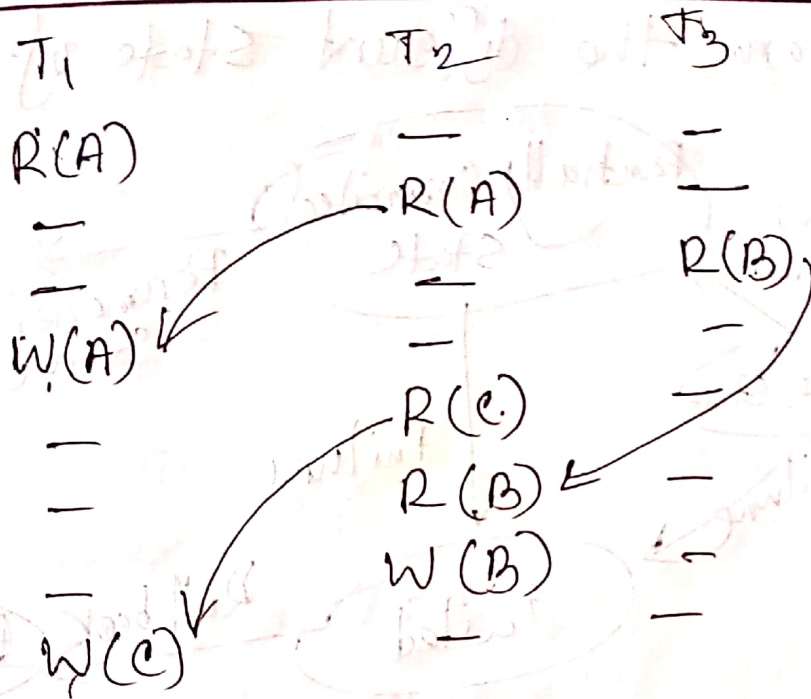
$A=0 \vee B=0 \Rightarrow F \vee T \Rightarrow T$

c)

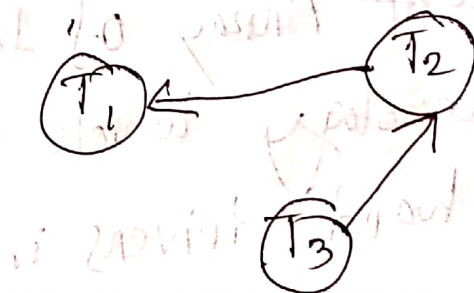
Check whether the schedule is conflict serializable or not?

S: $R_1(A); R_2(A); R_3(B); W_1(A); R_1(C);$
 $R_2(B); W_2(B); W_1(C).$

Schedule table:



T_2 conflict with T_1
 T_3 T_2
 T_2 T_1

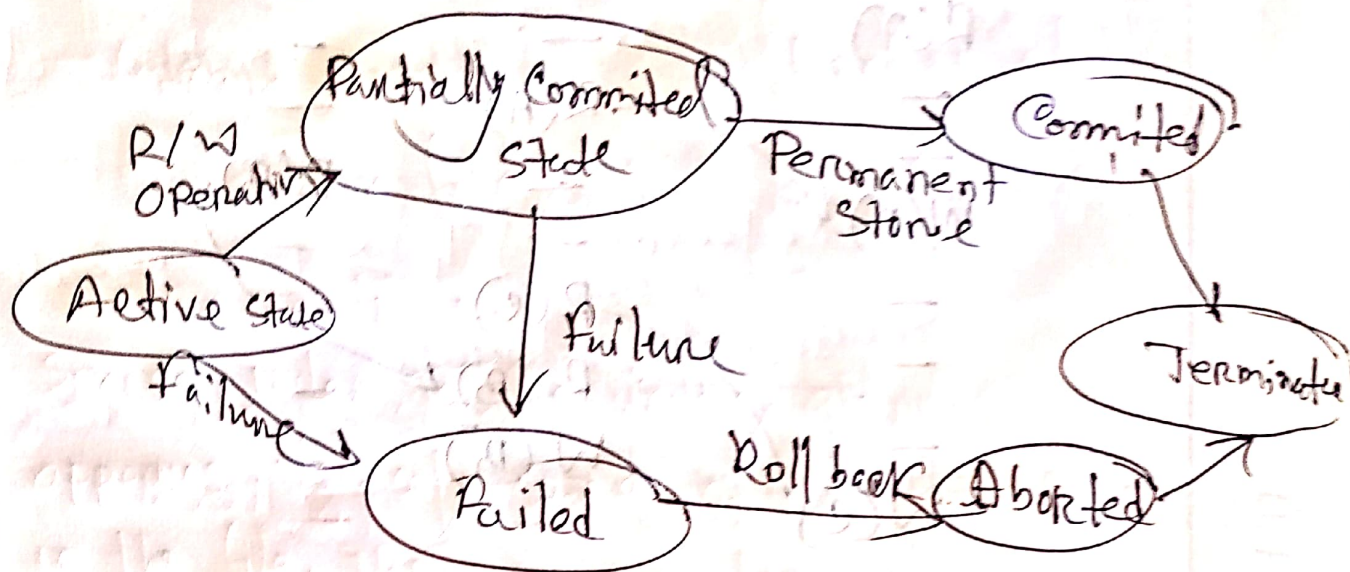


Precedence graph

No cycle has created so it is conflict serializable schedule. And order of Serializability:

~~$T_1 \rightarrow T_3 \rightarrow T_2$~~ $T_3 \rightarrow T_2 \rightarrow T_1$

Q) What are the different state of transaction



Q. a) Discuss how Raid can be used to improve reliability via redundancy.

RAID (Redundant Array of Independent Disks) is a technology which makes use of two or more hard drives in order to improve performance, reliability or create large data volumes.

i.e. The chance that at least one disk out of a set of N disks will fail is much higher than the chance that a specific disk

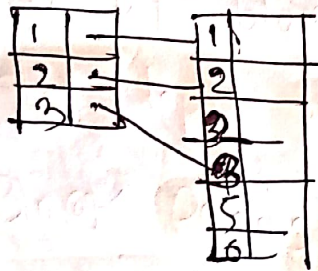
will fail)

ii. Redundancy stores extra information that is not needed normally, but that can be used in the event of failure of disk to rebuild the lost information.



b) What is Sparse Indexing? How does multilevel indexing improve the efficiency of searching an index file?

Sparse index is the index where an index record appears for only some of the search-key values in the file. They require less space & impose less maintenance overhead for insertions & deletions. Index entry for each page of data file.



multilevel index: An index with two or more levels is called a multi-level index. Searching of ^{records with a} multilevel index requires significantly fewer I/O operations than does searching for records by binary search.

Advantages of / how does multilevel indexing improve the efficiency of searching an index file:

- i. faster access to specific data
- ii. Reduced search time
- iii. Improved Data Management
- iv. Scalability:

Q A multilevel index reduces the number of blocks accessed when searching for a record, find its indexing field value

Q To retain the benefits of using multilevel indexing while reducing index insertion & deletion problems..

Q Designers adopted a multilevel indexing that reserves some space in each of its blocks for inserting new entries.

(1) $Block\ Size = (row) \times (col)$

(1-1) $(1000) \times (10) =$

10000

Assumption:

- Each index level is stored in one disk block
- There is no overflow block containing additional index entries
- The desired entry is found at the last level of the index structure (leaf level)

Best Scenario:

Formula:

$$\text{Depth} = \log_F(1000)$$

The number of block reads in the best case is the depth on the index structure.

$$\begin{aligned}\text{Block Reads (Best Case)} &= \text{Depth} \\ &= \log_F(1000)\end{aligned}$$

Worst:

$$\begin{aligned}\text{Block Reads (Worst Case)} &= \text{Depth} \times (F - 1) \\ &= \log_F(1000) \times (F - 1)\end{aligned}$$

$(F-1)$ represents the num of blocks

Instance of Relations:

Course Relation —

Course name	Room	Instruction
CSF001	6001	XYZ
MATH002		

Enrollment Relation —

Course name	Stu name	Grade
CSF001	X	A
CSF001	Y	A+
CSF001	Z	A+
MATH002		
MATH002		

File Structure with Clustering:

Block-1 —

Course name	Student name	Grade
CSF101	X	A
CSF101	Y	A+
CSF101	Z	A+

Block2-

Course name	Stu name	Grade
MATH002	X	B
MATH002	Y	A
MATH002	Z	B+

In file structure, records are physically ordered based on the "Course name".

records for the same course are clustered together, making it more efficient to retrieve information related to a specific course.

10001 10002 10003 10004 10005 10006 10007 10008 10009 10010 10011 10012 10013 10014 10015 10016 10017 10018 10019 10020 10021 10022 10023 10024 10025 10026 10027 10028 10029 10030 10031 10032 10033 10034 10035 10036 10037 10038 10039 10040 10041 10042 10043 10044 10045 10046 10047 10048 10049 10050 10051 10052 10053 10054 10055 10056 10057 10058 10059 10060 10061 10062 10063 10064 10065 10066 10067 10068 10069 10070 10071 10072 10073 10074 10075 10076 10077 10078 10079 10080 10081 10082 10083 10084 10085 10086 10087 10088 10089 10090 10091 10092 10093 10094 10095 10096 10097 10098 10099 10100

10101 10102 10103 10104 10105 10106 10107 10108 10109 10110 10111 10112 10113 10114 10115 10116 10117 10118 10119 10120 10121 10122 10123 10124 10125 10126 10127 10128 10129 10130 10131 10132 10133 10134 10135 10136 10137 10138 10139 10140 10141 10142 10143 10144 10145 10146 10147 10148 10149 10150 10151 10152 10153 10154 10155 10156 10157 10158 10159 10160 10161 10162 10163 10164 10165 10166 10167 10168 10169 10170 10171 10172 10173 10174 10175 10176 10177 10178 10179 10180 10181 10182 10183 10184 10185 10186 10187 10188 10189 10190 10191 10192 10193 10194 10195 10196 10197 10198 10199 10200

10201 10202 10203 10204 10205 10206 10207 10208 10209 10210 10211 10212 10213 10214 10215 10216 10217 10218 10219 10220 10221 10222 10223 10224 10225 10226 10227 10228 10229 10230 10231 10232 10233 10234 10235 10236 10237 10238 10239 10240 10241 10242 10243 10244 10245 10246 10247 10248 10249 10250 10251 10252 10253 10254 10255 10256 10257 10258 10259 10260 10261 10262 10263 10264 10265 10266 10267 10268 10269 10270 10271 10272 10273 10274 10275 10276 10277 10278 10279 10280 10281 10282 10283 10284 10285 10286 10287 10288 10289 10290 10291 10292 10293 10294 10295 10296 10297 10298 10299 10300

10301 10302 10303 10304 10305 10306 10307 10308 10309 10310 10311 10312 10313 10314 10315 10316 10317 10318 10319 10320 10321 10322 10323 10324 10325 10326 10327 10328 10329 10330 10331 10332 10333 10334 10335 10336 10337 10338 10339 10340 10341 10342 10343 10344 10345 10346 10347 10348 10349 10350 10351 10352 10353 10354 10355 10356 10357 10358 10359 10360 10361 10362 10363 10364 10365 10366 10367 10368 10369 10370 10371 10372 10373 10374 10375 10376 10377 10378 10379 10380 10381 10382 10383 10384 10385 10386 10387 10388 10389 10390 10391 10392 10393 10394 10395 10396 10397 10398 10399 10400

10401 10402 10403 10404 10405 10406 10407 10408 10409 10410 10411 10412 10413 10414 10415 10416 10417 10418 10419 10420 10421 10422 10423 10424 10425 10426 10427 10428 10429 10430 10431 10432 10433 10434 10435 10436 10437 10438 10439 10440 10441 10442 10443 10444 10445 10446 10447 10448 10449 10450 10451 10452 10453 10454 10455 10456 10457 10458 10459 10460 10461 10462 10463 10464 10465 10466 10467 10468 10469 10470 10471 10472 10473 10474 10475 10476 10477 10478 10479 10480 10481 10482 10483 10484 10485 10486 10487 10488 10489 10490 10491 10492 10493 10494 10495 10496 10497 10498 10499 10500

10501 10502 10503 10504 10505 10506 10507 10508 10509 10510 10511 10512 10513 10514 10515 10516 10517 10518 10519 10520 10521 10522 10523 10524 10525 10526 10527 10528 10529 10530 10531 10532 10533 10534 10535 10536 10537 10538 10539 10540 10541 10542 10543 10544 10545 10546 10547 10548 10549 10550 10551 10552 10553 10554 10555 10556 10557 10558 10559 10560 10561 10562 10563 10564 10565 10566 10567 10568 10569 10570 10571 10572 10573 10574 10575 10576 10577 10578 10579 10580 10581 10582 10583 10584 10585 10586 10587 10588 10589 10590 10591 10592 10593 10594 10595 10596 10597 10598 10599 10600