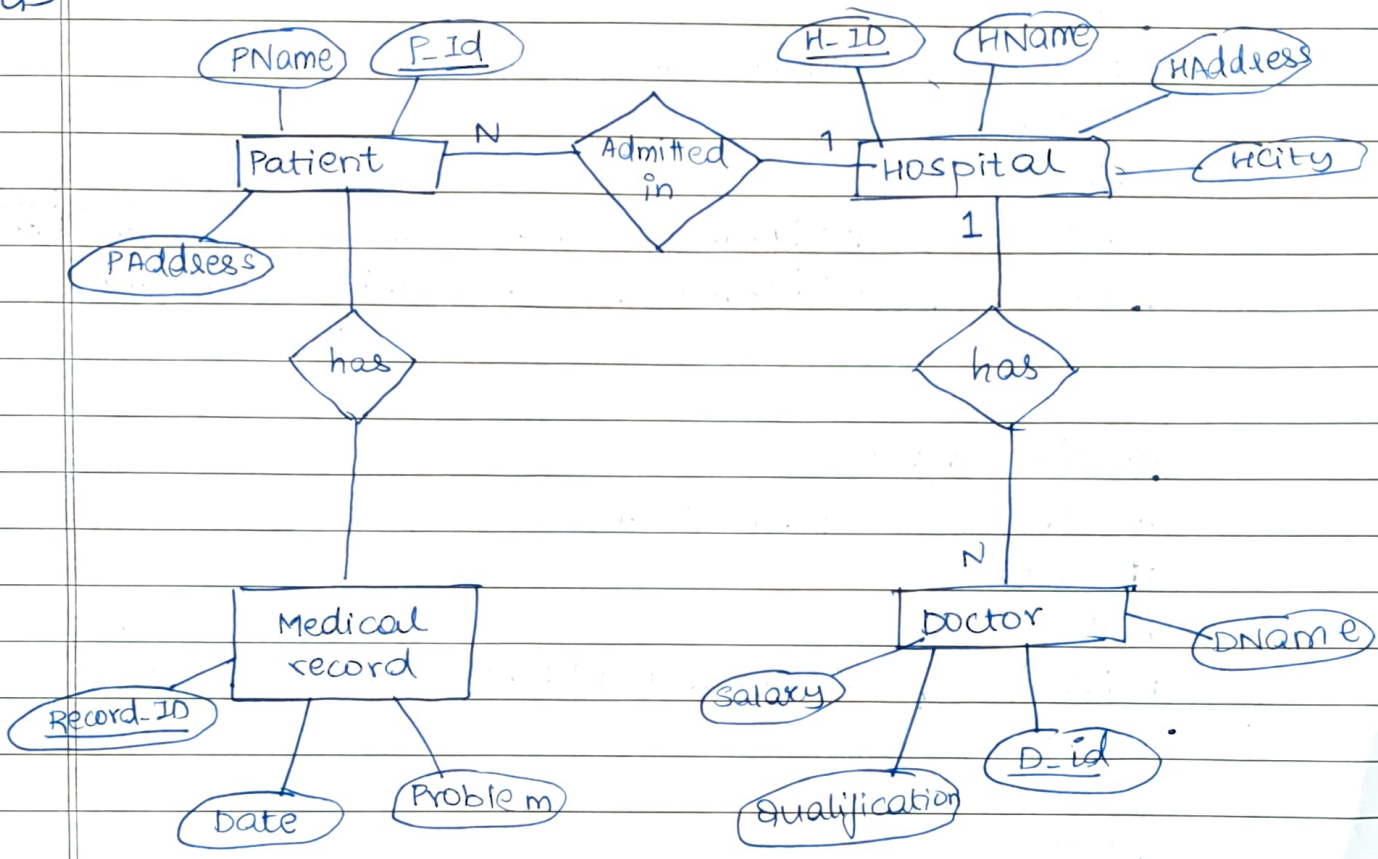
14.12.2020

END SEM-3 EXAM
DBMS

JUNAID. GIRKAR
6000419005S7
JAGirkar

Q3



RELATIONAL MODEL.

Hospital	P
H_id	Primary Key
H- Name	
HAddress	
Hcity	
P_id	Foreign Key references P_id of patient
D_id	Foreign Key references D_id of Doctor

JUNAID . GIRKARI
G000419005 57
JAGirkar

Patient	
P_id	Primary Key
PName	
PAddress	
Record_id	Foreign Key references Record_id of ~~medicine~~ medical record
H_id	Foreign key references H-id of Hospital

Medical Record	
Record_id	Primary Key
Problem	
Date	
P_id	Foreign key references P_id of patient

Doctor	
D-id	Primary Key
DName	
Qualification	
Salary	
H-id	Foreign key references H-id of hospital

JUNAID. GIRKAR
G000419005-7
JAGirkar

Q4

ANS (i)

```
SELECT  count (distinct name)
FROM  accident , participated , person
WHERE  accident.report number = participated.report number
AND  participated.driver id = person.driver id
AND  date  BETWEEN '01-01-2019' AND '31-12-2019'
```

(ii)

```
SELECT  count (distinct report number)
FROM  accident NATURAL JOIN participated
NATURAL JOIN person
WHERE  name = 'Sunil K.'
```

(iii)

```
UPDATE  participated
SET  damage amount = 5000
WHERE license = 'MUM2022' AND report number = 'AR2197'
```

PS: Underscore is not used for variable names in question and thus I have also not used in the solution

JUNAID. GIRKAR
G00041900057
JAGirkar

Q5

ANS In a database, a deadlock is an unwanted condition in which two or more transactions are waiting indefinitely for one another to give up locks. Deadlock is said to be one of the most feared complications in DBMS as it brings the whole system to a Halt.

For Example: consider the partial schedule

T_3	T_4
lock-x (B)	
read (B)	
B := B-50	
write (B)	
	lock-S (A)
	read (A)
	lock-S (B)
lock-x (A)	

Neither T_3 nor T_4 can make progress - executing lock-S(B) causes T_4 to wait for T_3 to release its lock on B, while executing lock-X (A) causes T_3 to wait for T_4 to release its lock on A. This situation is called a deadlock. System is deadlocked if there is a set of transactions such that every transaction in the set is waiting for another transaction in the same set.

JUNAID. GIRKAR
600041900 57
- JAGiKas

DEADLOCK PREVENTION

- Deadlock prevention protocols ensure that the system will never enter into a deadlock state. Some prevention strategies are :-
  - → Require that each transaction locks all its data items before it begins execution [PREDECLARATION]
  - → Impose partial ordering of all data items and require that a transaction can lock data items only in the order specified by the partial order.

- → Following schemas use transaction timestamps for the sake of deadlock prevention
  - wait-die scheme — non-preemptive.
    A transaction may die several times before acquiring needed data item.
  - wound-wait scheme - preemptive
    The wound-wait system does the opposite. A younger transaction is allowed to wait for an older one

- → Timeout-based schemes :-
  - A transaction waits for a lock only for a specified amount of time. If the lock has not been granted within that time, the transaction is rolled back and restarted.
  - This is simple to implement but starvation is possible

JUNAID - GIRKAR
6000419005
JAGirkar

5.2 DEADLOCK DETECTION

→ Deadlocks can be described as a wait-for graph, which consists of a pair $G = (V, E)$.

- V is a set of vertices (all the transactions in the system)
- E is a set of edges; each element is an ordered pair $T_i \rightarrow T_j$.
- one node is created in the wait-for graph for each transaction that is currently executing.
- whenever a transaction T_i is waiting to lock an item X that is currently locked by a transaction T_j, a directed edge ($T_i \rightarrow T_j$) is created in the wait-for graph.
- when T_j releases the lock(s) on the items that T_i was waiting for, the directed edge is dropped from the wait-for graph.
- we have a state of deadlocks if and only if the wait-for graph has a cycle

Q6

a

ANS In any organisation where many people use the same resources, there is a need for a chief administrator to oversee and manage the resources. In a database environment, the primary resource is the database itself and secondary are DBMS and related software. Administering these resources is the responsibility

of the database administrator (DBA). The DBA is responsible for authorizing access to the database, co-ordinating and monitoring its use and acquiring software and hardware resources as needed.

The main functions of the DBA are :-

- SCHEMA DEFINATION: The DBA creates the original database schema.
- STORAGE STRUCTURE & ACCESS-METHOD DEFINATION: The DBA creates appropriate storage structures and access methods by writing a set of definations.
- SCHEMA AND PHYSICAL-organisation modification
- Granting of authorization for data access
- Integrity-constraint specification.

6b

ANS Fragmentation is a database server feature that allows you to control where data is stored at the table level. Fragmentation enables you to define groups of rows or index keys within a table according to some algorithm or scheme. You can store each group or fragment/partition in a seperate dbspace associated with a ~~physical~~ specific physical disk. You use SQL statements to create the fragments and assign them to dbspaces.

Information about the fragmentation of the data is stored in DDC.

JUNAID. GIRKAR
6000419005 9
JAGirkar

6b

fragmentation of data is the task of dividing a table into a set of smaller tables. The subsets are called fragments.

Fragmentation of databases is required if the goal is :-
- single-use response time
- concurrency
- loading of data
- Availability
- Back up and restore characteristics

• Fragmentation of database allows :
- Easy use of data
- Query efficiency
- Reliability of fetched data
- Evenly balanced storage distribution

ORIGINAL TABLE

id	Name	sex	Age
1	Harry	M	5
2	Ron	M	6
3	Hermoine	F	7
4	Sirius	M	8

HORIZONTAL FRAGMENTATION OF ABOVE TABLE

id	Name	sex	Age
1	Harry	M	5
2	Ron	M	6
3	Hermoine	F	7
4	Sirius	M	8

Types of Horizontal Fragmentations:

- PHF [Primary Horizontal Fragmentation]
- DHF [Delived Horizontal Fragmentation]

Types of Fragmentation
- Vertical Fragmentation
- Horizontal Fragmentation

Vertical Fragmentation of original table

id	Name
1	Hally
2	Ron
3	Hermoine
4	Sirius

id	sex
1	M
2	M
3	F
4	M

id	Name	sex	Age
1	Hally	M	5
2	Ron	M	6
3	Hermoine	F	7
4	Sirius	M	8