APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FOURTH SEMESTER B.TECH DEGREE EXAMINATION, June 2017

Branch: **Computer Science and Engineering**

CS208: Principles of Database Design

ANSWER KEY & SCHEME OF VALUATION

1. *List out any three salient features of database systems* (3)

List any three of the following features **1 mark/feature**

- self –describing
- insulation of data and programs (also known as program-data independence
- data abstraction
- support for multiple views
- multi-user (concurrent) transaction processing.

2. *How is DML different from DDL? Write a sample statement in DML and one in DDL.* (3)

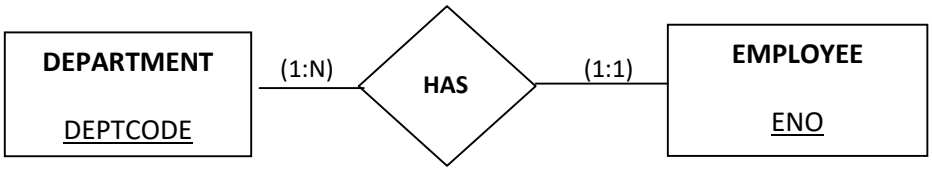
two- to-three sentence explanation of DML and DDL highlighting the difference **1 mark**

DDL (data definition language) is used to describe the structures that hold data in the database; DML (data manipulation language) is used to express query and update operations on the data stored in the database.

A sample DDL statement like CREATE TABLE (...), ALTER TABLE …, DROP TABLE with sample table names, attribute names etc. Correct syntax need not be insisted completely. **1 mark**

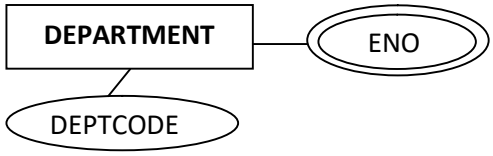
A sample DML statement like SELECT…, INSERT…, DELETE… etc. with some sample table names, attribute names etc. Correct syntax need not be insisted completely. **1 mark**

3. *Can we represent the situation modelled by the following ER diagram without the relationship 'HAS'? If so, draw the new diagram. If not, give the reasons. (Entities are DEPARTMENT and EMPLOYEE. Attibutes names are given under entity names; keys are underlined.)* (3)



The intention is to test the student's unberstanding of multi-valued attribute. (The attributes DNAME and ENAME are some how missing in the QP. Still the question can be answered correctly.)

Here the EMPLOYEE can be added as a multi-valued attribute of DEPARTMENT as follows:



If the student a similar ER diagram with correct noation 3 marks
Notation is not correct but mentions that ENO is a multi-valued attribute 2.5 marks
Notation is not correct but mentions that ENO is an attribute 1.5 marks
Notation is correct but mentions that ENO is an attribute 2 marks
(Student may also show DEPARTMENT as a mutli-valued attribute of EMPLOYEE. This is incorrect considering the cardinality and participation constraints given.)

4. *Consider the a relation R(A,B,C,D) where A is a key of R. Write any three relational algebra expressions equivalent to* $\prod_{A,B}(\sigma_{A=2 \text{ and } B=3}(R))$ (3)

The question aims to test the student's understanding of associative and commutative nature of some RA operators. In particular, note that since A is a key of R, even $\prod$ and σ can be commuted. Some of the equivalent expressions are,

- $\prod_{A,B}(\sigma_{A=2}(\sigma_{B=3}(R)))$
- $\prod_{A,B}(\sigma_{B=3}(\sigma_{A=2}(R)))$
- $\prod_{A}(\prod_{B}(\sigma_{B=3}(\sigma_{A=2}(R))))$
- $\sigma_{A=2}(\sigma_{B=3}(\prod_{A,B}(R)))$
- $\sigma_{A=2 \text{ and } B=3}(\prod_{A,B}(R))$
- $\sigma_{A=2 \text{ and } B=3}(\prod_{A}(\prod_{B}(R)))$

1 mark/ equivalent expression = 3 marks

PART B

Answer any two full questions

5. *Study the tables given below and write relational algebra expressions for the queries that follow.* (9)
STUDENT(ROLLNO, NAME, AGE, GENDER, ADDRESS, ADVISOR)
COURSE(COURSEID, CNAME, CREDITS)
PROFESSOR(PROFID,PNAME, PHONE)
ENROLLMENT(ROLLNO, COURSEID, GRADE)
Primary keys are underlined. ADVISOR is a foreign key referring to PROFESSOR table. ROLLNO and COURSEID in ENROLLMENT are also foreign keys referring to THE primary keys with the same name.

(i) Names of female students
(ii) Names of male students along with adviser name
(iii) Roll Number and name of students who have not enrolled for any course. This query can be answered using the difference operator in multiple-steps As follows:

(i) $\prod_{\text{NAME}}(\sigma_{\text{GENEDER='F...'}}(\text{STUDENT}))$ **2 marks**

(ii) $\prod_{NAME,\ PNAME}(STUDENT \underset{ADVISOR=PROFID}{\bowtie} PROFESSOR)$ **3 marks**

This query can be answered using the difference operator in multiple-steps as follows:

ROLLNO-NOT-ENROLLED(ROLLNO)$\leftarrow \prod_{ROLLNO}(STUDENT) - \prod_{ROLLNO}(ENROLLMENT)$

STUDENT-NOT-ENROLLED(ROLLNO,NAME)$\leftarrow \prod_{ROLLNO,NAME}$ (ROLLNO-NOT-ENROLLED*STUDENT)

In a single step the query can be,

$\prod_{ROLLNO,NAME}((\prod_{ROLLNO}(ENROLLMENT) - \prod_{ROLLNO}(ENROLLMENT))*STUDENT)$

4 marks

Some students may also answer as single expression using the relation-rename-operator, ρ (as followed in Korth et.al.), and can be given full credit.

Equivalent expressions that give correct answers can be given marks.

6. *A company has the following scenario: There are a set of salespersons. Some of them manage other salespersons. However, a salesperson cannot have more than one manager. A salesperson can be an agent for many customers. A customer is managed by exactly one salesperson. A customer can place any number of orders. An order can be placed by exactly one customer. Each order lists one or more items. An item may be listed in many orders. An item is assembled from different parts and parts can be common for many items. One or more employees assemble an item from parts. A supplier can supply different parts in certain quantities. A part may be supplied by different suppliers.* (9)

(i) Identify and list entities, suitable attributes, primary keys, foreign keys and relationships to represent the scenario.

(ii) Draw an ER diagram to model the scenario using min-max notation.

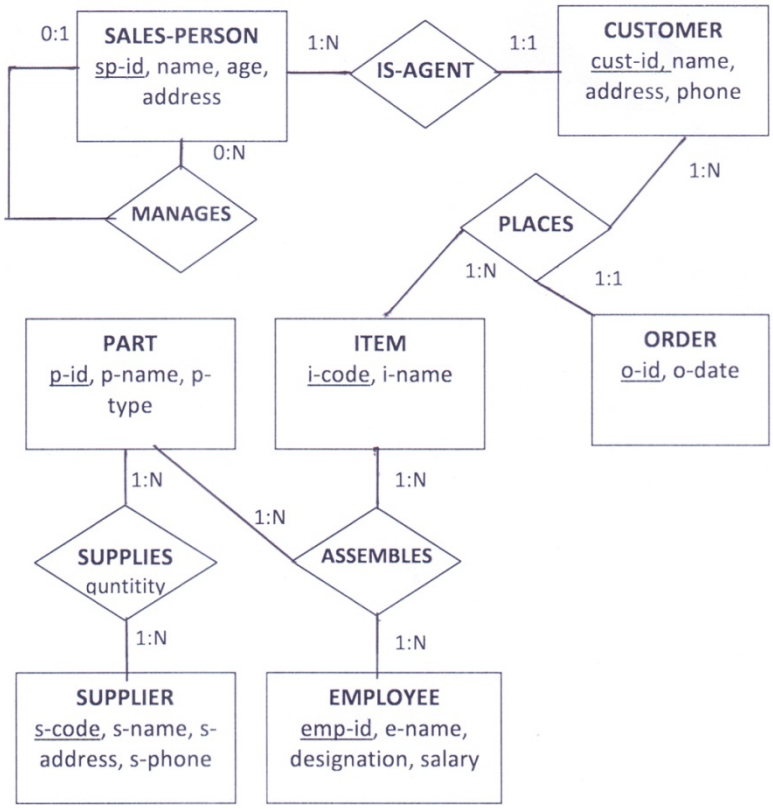
(i) ('Foreign Key' is a mistake and can be ignored.)

Entity sets Entities = 2 marks; Relationships = 1 mark; suitable attributes = 1 mark; keys = 1 mark;

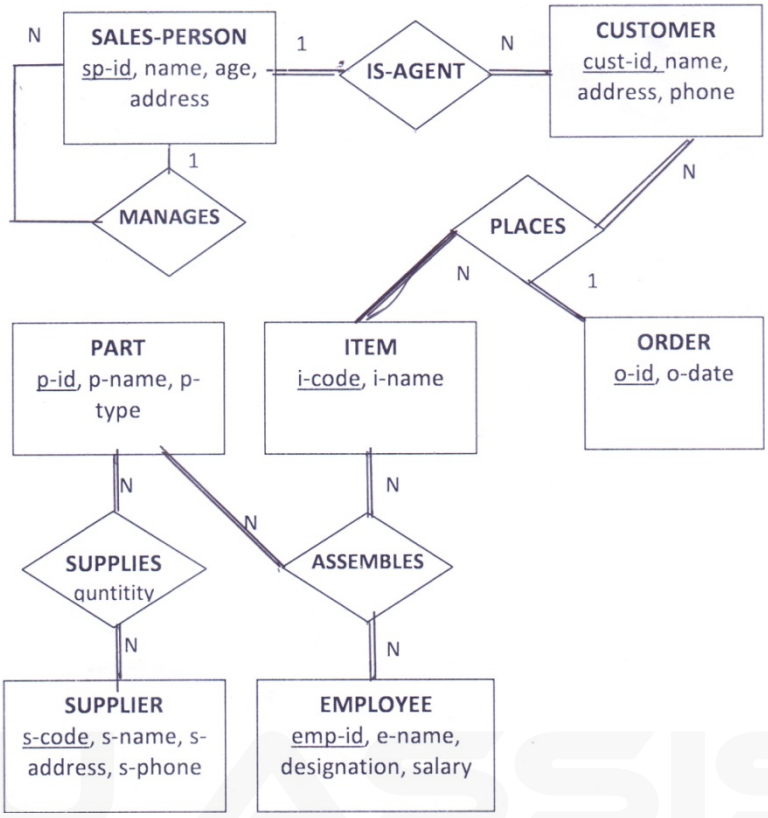
(ii) A representative ER diagram (using a non-standard notation) is shown below. The student is expected to draw an diagram representing the facts using correct notation for entities, attributes, keys, etc. and participation and cardinality constraints. Credits can be given even if the (min:max) notation is not used, if he/she shows the cardinality and participation constraints correctly using conventional notation.

4 marks

B4E561



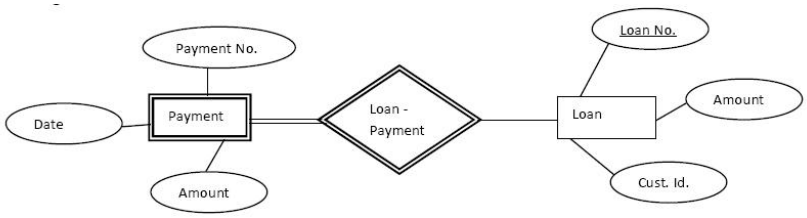
Using Conventional Notation for cardinality and participation constraints.



7. a. *Justify the importance of weak entity sets with the help of an example.* *(3)*

A diagram showing a weak entity set, identifying relationship and owner entity set is expected. In addition an explanation on why weak entity set is important in the given situation also has to be given.

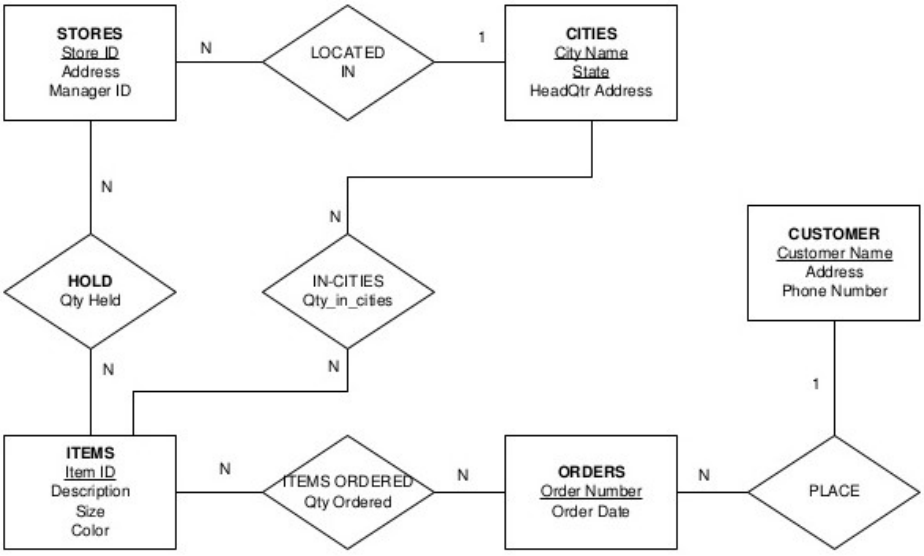
Sample:



The ER diagram represents payment made against each loan. As different loans can have the same payment number (i.e. instalment number), 'Payment No.' cannot be a key of the 'Payment' entity set and hence it is weak entity set connected to the strong entity set 'Loan'.

Diagram : 1.5 marks
Explanation: 1.5 marks

b. *In the ER diagram below, names of entity sets and relationships are shown in capital and corresponding attributes are listed under each such name. Key attributes are underlined. All the participations are total. Use the standard synthesis procedure to convert the ER diagram into the corresponding relational schema. Clearly show primary and foreign keys.* *(6)*



Tables (corr. to entity sets): **4.5 marks**

STORES(StoreID, Address, ManagerID, *CityName,State*)
CITIES(CityName,State, HeadQtrAddress)
ITEMS(ItemID, Description, Size, Color)
ORDERS(OrderNumber, OrderDate, CustomerName)
CUSTOMER(CustomerName, Address, Phone Number)
(here primary keys are underlined; foreign keys are double underlined)

Tables (corr. To relationship sets) **1. 5 marks**

IN-CITIES(Item ID, City Name, State, Qty-in-Cities)
HOLD(store ID, Item ID, QtyHeld)
ITEMS-ORDERED(item ID, Order Number, Qty-Ordered)
(here primary keys, which have foreign keys as components, are double underlined)

Full marks ***only*** if all the tables are shown with all the required attributes and if the foreign and primary keys are clearly mentioned.

PART C
Answer all questions

8. *Illustrate the GROUP BY clause with the help of a real example.* *(3)*

One/two sentence explanation of GROUPBY **1 mark**

A sample table, Correct SQL expression with GROUP BY and output **2 marks**

Note that the attributes appearing in GROUP BY clause should appear in SELECT clause also.

9. *Consider the query SELECT NAME, AGE FROM STUDENT WHERE GENDER = 'Male' on the table STUDENT(ROLLNO, NAME, AGE, GENDER, ADDRESS). Give a relational algebra expression corresponding to the query. Is result produced by the query and your expression always the same? Why?* *(3)*

RA Expression: $\prod_{NAME, AGE} (\sigma_{GENDER='Male'} (STUDENT))$ (or equivalent)

2 marks

RA eliminates duplicates. Thus the result need not be the same. In case NAME, AGE or both together form a key of the STUDENT table, result of the query and the RA expression will always be the same. **1 mark**

10. *Determine any two candidate keys of the relation R(A,B,C,D,E,F) with FDs AB→C,C→AD, D→EF, F→B.* *(3)*

Determining each key **1.5 marks**

To determine the keys, student can use the standard algorithm to compute X+; alterntively, he/she can use Amstrongs Axioms. The steps should be shown for giving full marks.

AB and C are the two obivious keys. But students may come up with other keys through Armstrong' Axioms. Make sure that they are minimal super keys before awarding marks.

11. *Give an example for a relation that has insertion, deletion and update anomalies. Which type(s) of functional dependency can formally model these anomalies? Quote one such dependency from your example* *(3)*

Any relation which is a result of combining two real-world entity sets will have these anomalies.

The student us expected to

- Give one such relation. For example, STUDENT-COURSE(ROLLNO, COURSEID, NAME, CLASS, CNAME, CCREDIT, GRADE). **1 mark**
- Two types of the dependencies that model it – transitive dependency and partial functional dependency. **1mark**
- Quote any one of PFD or TFD from the given relation. (Example: {ROLLNO, COURSEID } → CNAME while COURSEID → CNAME is PFD) **1 mark**

PART D

Answer any two full questions

12. a. *Illustrate the use of assertions with a typical example.* *(3)*

A typical simple assertion with brief explanation is expected.

Assertions are used to specify general restrictions on data stored in tables. These restrictions cannot be expressed using integrity constraints.
Example:

```
CREAT ASSERTION SALARY_CONSTRAINT
CHECK (NOT EXISTS (SELECT *
          FROM EMPLOYEE E, EMPLOYEE M,
                    DEPARTMENT D
          WHERE E.SALARY > M.SALARY AND
               E.DNO=D.NUMBER AND
                    D.MGRSSN=M.SSN))
```

The above assertion makes sure that salary of an employee does not exceed that of his/her

manager. The assertion part comes within the *check* clause. For every update on salary of the employee, the database checks the condition given by the assertion and alarms if it fails.

example: 1.5 marks, explanation: 1.5 marks

b. *Consider a relation (A,B,C,D,E,F) with A as the only key. Assume that the dependencies E→F and C→DEH hold on R.* (6)

(i) Is R is in 2NF? If not, decompose to 2NF.

(ii) Is R is in 3NF? If not, decompose to 3NF.

(Here the attribute H is missing. However that does not affect the approach to answer the question. The presence of H is quite irrelevant as it does not appear on the left side of any FD. C→DEH and be taken as C→DE.)

(i) There is no partial functional dependency as the only key is *not* composite. Hence the relation in 2NF. **2marks**

(ii) There are two transitive dependencies – through E→F and through C→DE. Therefore decompose the original relation into,

1. R1(A,B,C,E)
2. R2(C,D,E)
3. R3(E,F)

Primary keys underlined; foreign keys double-underlined.

Complete decomposition and a description similar to the above: **4 marks**

13. *In the following tables ADVISOR and TAUGHTBY are foreign keys referring to the table PROFESSOR. ROLLNO and COURSEID in ENROLLMENT refer to tables with primary keys of the same name.* (9)

STUDENT(ROLLNO, NAME, AGE, GENDER, ADDRESS, ADVISOR)
COURSE(COURSEID, CNAME, TAUGHTBY, CREDITS)
PROFESSOR(PROFID,PNAME, PHONE)
ENROLLMENT(ROLLNO, COURSEID, GRADE)

Write SQL expressions for the following queries:

(i) Names of courses taught by 'Prof. Raju'.

(ii) Names of students who have not enrolled for any course taught by 'Prof. Ganapathy'.

(iii) For each course, name of the course and number of students enrolled for the course.

Each of the above queries can be expressed in multiple ways in SQL. All syntactically and semantically correct expressions can be awarded marks. The most efficient ones are shown below:

(i) SELECT C.CNAME FROM PROFESSOR P, COURSE C WHERE P.PROFID = C.TAUGHTBY AND P.PNAME = 'Prof. Raju' **2 marks**

(ii) Being a negation query, the best way to express is through nested query.
SELECT S.NAME FROM STUDENT S WHERE S.ROLLNO NOT IN (SELECT E.ROLLNO FROM ENROLLMENT E, COURSE C, PROFESSOR P WHERE E.COURSEID=C.COURSEID AND C.TAUGHTBY = P.PROFID AND P.PNAME = 'Prof. Ganapathy')

Instead of 'NOT IN', 'NOT = ANY' can also be used. **4 marks**

(iii) SELECT C.CNAME COUNT (*) FROM COURSE C, ENROLLMENT E WHERE C.COURSEID=E.COURSEID GROUP BY CNAME
or
SELECT CNAME COUNT (*) FROM COURSE NATURAL JOIN ENROLLMENT GROUP BY CNAME
or
SELECT C.CNAME COUNT (*) FROM COURSE C JOIN ENROLLMENT E ON C.COURSEID=E.COURSEID GROUP BY CNAME

3 marks

(Instead of COUNT(*), COUNT(E.ROLLNO) can also be used. Note that the SELECT clause should contain the attribute appearing in the GROUP BY clause. The query assumes that every course is enrolled by at least one student. If this assumption is lifted the query can be answered OUTER JOIN. However, OUTER JOIN is not part of the syllabus. Nevertheless, if a student answers correctly using OUTER JOIN marks can be awarded.

SELECT C.CNAME COUNT (*) FROM COURSE C LEFT OUTER JOIN ENROLLMENT E ON C.COURSEID=E.COURSEID GROUP BY CNAME)

14. *Assume that the relation R(P,Q,S,T,U) with FDs P → S, Q → S, S → T, TU → S, SU → Pis decomposed into 5 relations: R1(P,T), R2(P,Q), R(Q,U), R4(S,T,U) and R5(P,U). Apply the standard algorithm to test if the decomposition isalossless-join decomposition.* (9)

The student is expected to start with an initial table similar to the one below and show the status after applying each FD. (Note:- There are can be many matrix sequences, depending on the order in which the dependencies are applied. Some of them will take more number of iterations to converge and some less. The sequence when the FDs are applied in the given order is shown below.)

Initial table:

	P	Q	S	T	U
R1	a_1	b_{12}	b_{13}	a_4	b_{15}
R2	a_1	a_2	b_{23}	b_{24}	b_{25}
R	b_{31}	a_2	b_{33}	b_{34}	a_5
R4	b_{41}	b_{42}	a_3	a_4	a_5
R5	a_1	b_{52}	b_{53}	b_{54}	a_5

P → S

	P	Q	S	T	U
R1	a_1	b_{12}	b_{13}	a_4	b_{15}
R2	a_1	a_2	~~b_{23}~~ b_{13}	b_{24}	b_{25}
R	b_{31}	a_2	b_{33}	b_{34}	a_5
R4	b_{41}	b_{42}	a_3	a_4	a_5
R5	a_1	b_{52}	~~b_{53}~~ b_{13}	b_{54}	a_5

Q → S

	P	Q	S	T	U
R1	a_1	b_{12}	b_{13}	a_4	b_{15}
R2	a_1	a_2	~~b_{23}~~ b_{13}	b_{24}	b_{25}
R	b_{31}	a_2	~~b_{33}~~b_{13}	b_{34}	a_5
R4	b_{41}	b_{42}	a_3	a_4	a_5
R5	a_1	b_{52}	~~b_{53}~~ b_{13}	b_{54}	a_5

S → T

	P	Q	S	T	U
R1	a_1	b_{12}	b_{13}	a_4	b_{15}
R2	a_1	a_2	~~b_{23}~~ b_{13}	~~b_{24}~~a_4	b_{25}
R	b_{31}	a_2	~~b_{33}~~b_{13}	~~b_{34}~~a_4	a_5
R4	b_{41}	b_{42}	a_3	a_4	a_5
R5	a_1	b_{52}	~~b_{53}~~ b_{13}	~~b_{54}~~a_4	a_5

TU → S

	P	Q	S	T	U
R1	a_1	b_{12}	b_{13}	a_4	b_{15}
R2	a_1	a_2	~~b_{23}~~ b_{13}	~~b_{24}~~a_4	b_{25}
R	b_{31}	a_2	~~b_{33}~~ ~~b_{13}~~ a_3	~~b_{34}~~a_4	a_5
R4	b_{41}	b_{42}	a_3	a_4	a_5
R5	a_1	b_{52}	~~b_{53}~~ ~~b_{13}~~ a_3	~~b_{54}~~a_4	a_5

SU → P

	P	Q	S	T	U
R1	a_1	b_{12}	b_{13}	a_4	b_{15}
R2	a_1	a_2	~~b_{23}~~ b_{13}	~~b_{24}~~a_4	b_{25}
R	~~b_{31}~~ a_1	a_2	~~b_{33}~~ ~~b_{13}~~ a_3	~~b_{34}~~a_4	a_5
R4	~~b_{41}~~ a_1	b_{42}	a_3	a_4	a_5
R5	a_1	b_{52}	~~b_{53}~~ ~~b_{13}~~ a_3	~~b_{54}~~a_4	a_5

As the third row is has all the cells with a-values, the decomposition is loss-less.

2marks/FD application step

Full credit should be given only if the matrix corresponding to processing each functional dependency is shown separately. Partial credits proportional to the extent up to which the computation is correct.

PART E

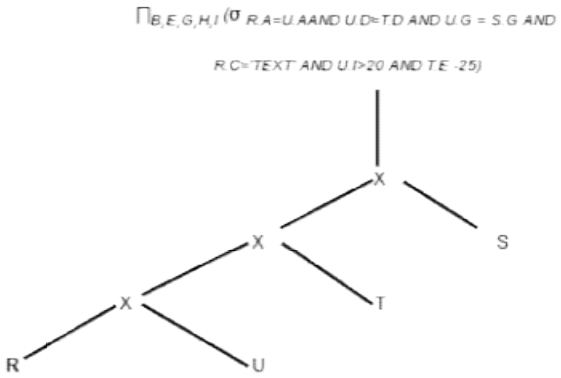
Answer any four full questions

15. *Consider the tables R (A, B, C), T(D,E,F), S(G, H) and U(A,D, G, I) where A, D and G in U are foreign keys referring to the primary keys with the same names. Show an initial query tree for the following query and optimize it using the rules of heuristics:* (10)

select B, E, G, H, I
from R, T, S, U

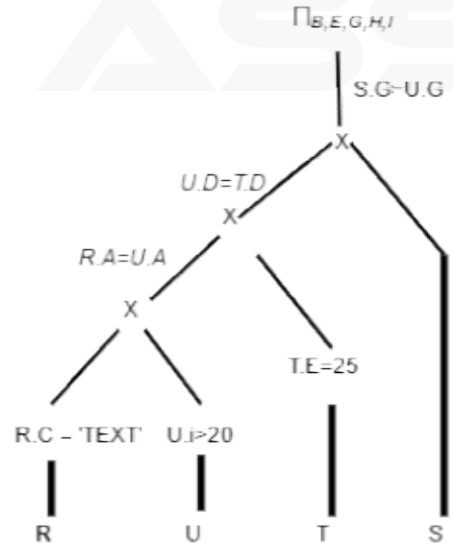
where R.A = U.A
and T.D = U.D and S.G = U.G
and R.C = 'TEXT' and U.I > 20 and T.E = 25

There are 4! initial, left-deep query trees, one for each possible order of the four tables R,T, S, U and the student can start with any one of them (or even with a tree which is right-deep or neither of them.) One such tree corresponding to the order R-U-T-S is shown below.



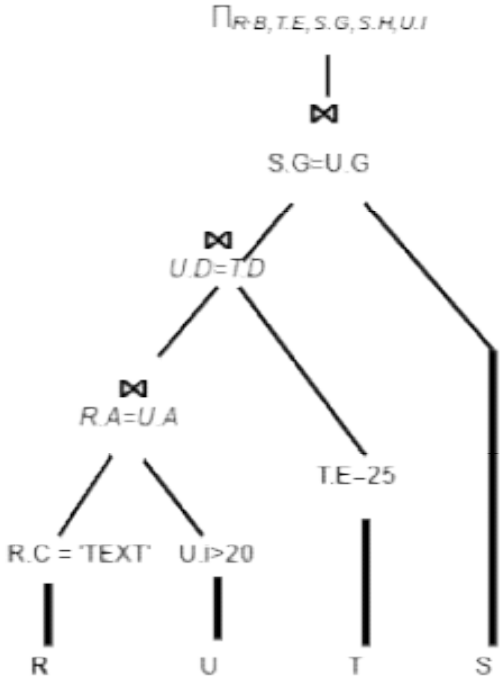
1 mark

After pushing down selection



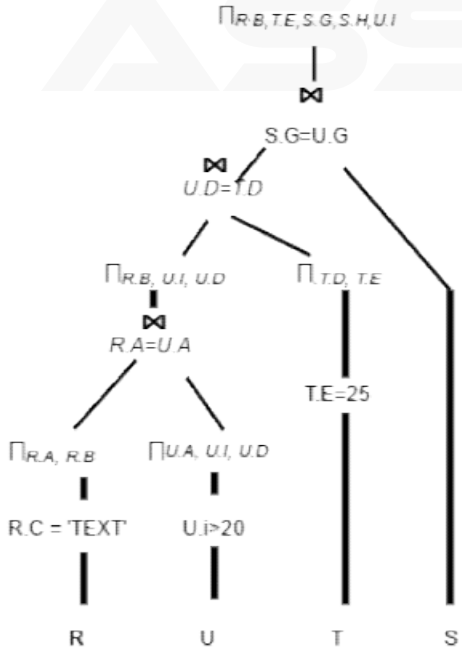
3 marks

After replacing cross-product-followed-by-selection by Join



3 marks

After pushing down projection



3 marks

If all the above steps are accurate, full credit can be given. If any intermediate step is missing, or some steps / final tree are incorrect, reduce marks proportionally.

16 *Consider a file with 2,00,000 records stored in a disk with fixed length blocks of size 256 bytes. Each record is of size 50 bytes. The primary key is 4 bytes and block pointer is 6 bytes. Compute the* (10)

following, assuming that multi-level primary index is used as access path:

(i) Blocking factor for data records
(ii) Blocking factor for index records
(iii) Number of data blocks
(iv) Number of First level index blocks
(v) Number of levels of multi level index

Given data :
Number of records (n)= 200000
Block size (B) = 256 bytes
Record size (r) = 50 bytes

(i) bfr_{data} = floor(256/50) = **5** **1 mark**

(ii) index record size = 6+4 = 10 bytes; bfr_{index} = floor(256/10) = **25** **1 mark**

(iii) Number of data blocks ceil(200000/5) = **40000** **1 mark**

(iv) Number of first level index records = number of data blocks = 40000
Number first level index blocks = ceil(40000/ bfr_{index}) = ceil(40000/25)= **1600**
2 marks

(v) Number of second level index records = number of first level index blocks = 1600
Number second level index blocks = ceil(1600/ bfr_{index}) = ceil(1600/25)= 64
Number of third level index records = number of first level index blocks = 64
Number third level index blocks = ceil(64/ bfr_{index}) = ceil(64/25)= 3
Number of 4th level index records = number of first level index blocks = 3
Number 4th level index blocks = ceil(3/ bfr_{index}) = ceil(3/25)= 1
Therefore, number of levels of multi-level index = 4 **5 marks**

50% credits may be given if the above computation steps are shown even though the final numerical values are not correct. Note that incorrect result in one step can lead to incorrect results in the subsequent steps.

17. a. *Argue that two-phase locking ensures serializability.* *(4)*

Only an argument of the correctness of 2PL with a supporting example is expected. Description of the 2PL protocol is not expected at all.

T1	T2
read_lock (Y);	read_lock (X);
read_item (Y);	read_item (X);
unlock (Y);	unlock (X);
write_lock (X);	Write_lock (Y);
read_item (X);	read_item (Y);
X:=X+Y;	Y:=X+Y;
write_item (X);	write_item (Y);
unlock (X);	unlock (Y);

In 2PL the locks granted in the growing phase and revoked in the shrinking phase. Therefore a transaction, say T1, is granted a lock, the other transaction say T2, has to wait until the locks are released. This leads to a scenario where the transactions are serialized in

the order of releasing locks.

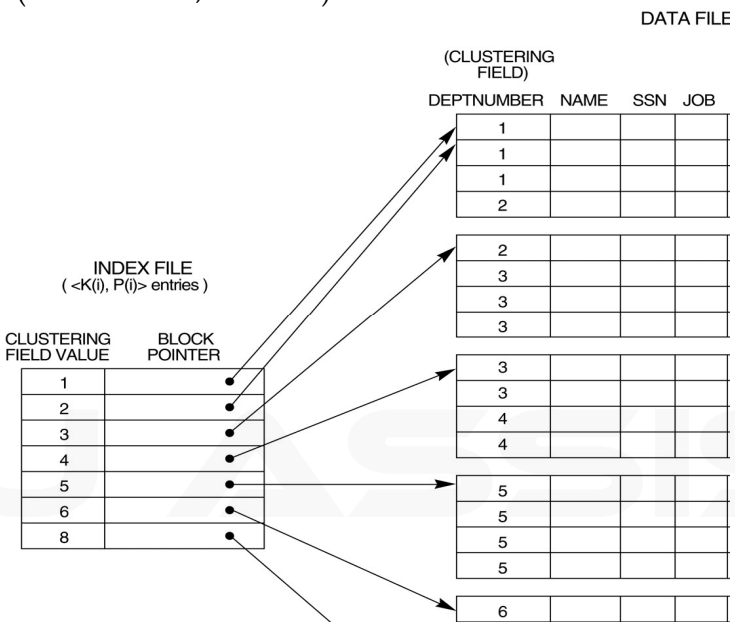
Argument (similar to above): 3 marks
Supporting diagram : 1 mark

b. *Illustrate clustering index and secondary index with typical, real examples.* *(6)*

Clustering index: a typical diagram showing how the index is organized and brief explanation with the following points:

- Clustering index is sparse index and is formed based on a non-key ordering field
- The pointer field of each index record points to the data block containing the first data record with the key value of the index record.

A typical diagram (from Elmasri, Navathe)



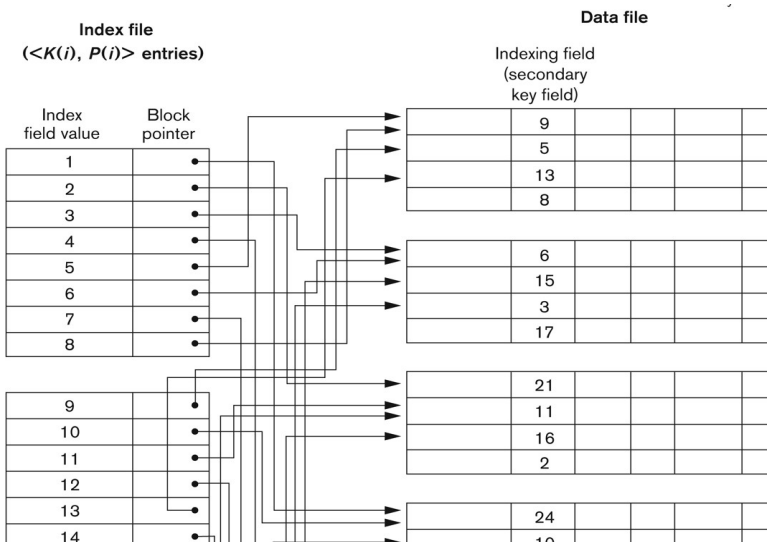
In the diagram DEPATMENT NUMBER is the non-key ordering filed and for each unique value there is block pointer.

diagram 1 mark, explanation: 2 marks

Secondary index: a typical diagram showing how the index is organized and brief explanation with the following points:

- Secondary index is dense index and is formed based on a non-ordering field.
- The pointer field of each index record contains the block address or record address of the record with the key value of the index record.

A typical diagram (from Elmasri, Navathe)

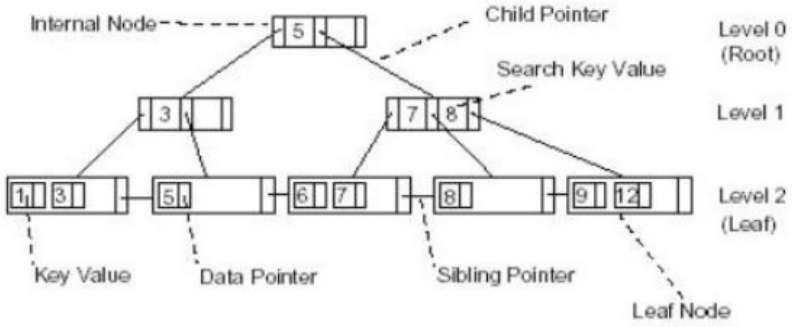


In the diagram there is an index record for every value of the index key is the non-key ordering filed and for each unique value there is block pointer.

diagram 1 mark, explanation: 2 marks

18. a. *Show the generic structure of a B+-Tree clearly indicating the types keys and pointers and their significance.* (5)

A diagram similar to the pne showing the types of nodes, node structure and pointer types and possibly some some sample values.



3 marks

A brief explanation of the way values are organized, with the following points, is expected:-

(i) There are n values and n+1 pointers in an internal node for some n

(ii) The first pointer points to the sub-tree containing values less than or equal to the first value.

(iii) kth pointer points to a sub-tree containing values less than or equal to the kth value and greater than (k-1)th value.

(iv) (n+1)th pointer points to sub-tree containing values greater than nth value.

2 marks

b. *What is the significance of check-pointing? Illustrate with a typical example.* (5)

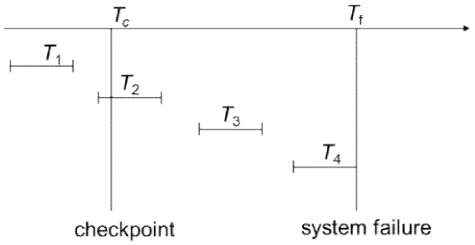
An explanation with the following points:

(i) Searching the entire log is time-consuming since we might unnecessarily redo transactions which have already and output their updates to the database. „

(ii) Check-pointing (a) outputs all log records currently residing in main memory onto stable storage (b) Output all modified buffer blocks to the disk (c) Write a log record < checkpoint> onto stable storage.

2 marks

A diagram similar to the following and brief explanation.



Scan backwards from end of log to find the most recent <check point> record and continue scanning backwards till a <Ti, start…>. Starting from Ti or later with <Ti, commit>, execute redo (Ti). (In the example, T2 and T3)

3 marks

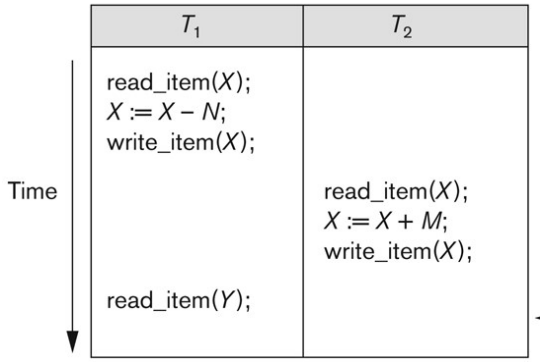
19. a. *Illustrate lost-update and dirty-read problems with suitable examples.* *(4)*

Lost update problem: The value updated by one transaction is over-written by another transaction and hence is lost. In the example below, value of X written by T1 is again updated by T2 and hence the original value is lost.

Time	T_1	T_2
↓	read_item(X); $X := X - N$;	
		read_item(X); $X := X + M$;
	write_item(X); read_item(Y);	
		write_item(X);
	$Y := Y + N$; write_item(Y);	

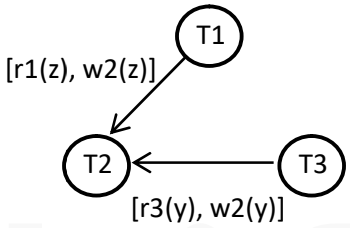
Explanation similar to the above – 1 mark
Example similar to the above – 1 mark

Dirty-read problem:- The value read by transaction is the one written by a transaction that has been rolled back after failure. In the example, if the transaction T1 writes X and fails at later stage and roll back, the value of X read by T2is a dirty value.



Explanation similar to the above – 1 mark
Example similar to the above – 1 mark

b. *Determine if the following schedule is serializable.* *(6)*
r1(X), r2(Z), r1(Z), r3(X), r3(Y), w1(X), w3(Y), r2(Y), w2(Z), w2(Y)
(Note: ri(X)/wi(X) means transaction Ti issues read/write on item X)



The schedule is serializable since there is not cycle in the *precedence graph*.

Full credit to be given, if the graph is shown with conflicting instructions as labels and state why the schedule is serializable. **6 marks**

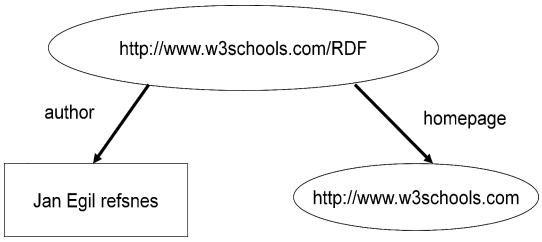
20. a. *Write a small RDF document and show its equivalent graph structure.*

A very simple RDF textual description as XML document or RDF triples and the corresponding graph structure is expected.
Sample:
XML

```
<RDF>
  <Description about="http://www.w3schools.com/RDF">
    <author>Jan Egil refsnes</author>
    <homepage>http://www.w3schools.com</homepage>
  </Description>
</RDF>
```

Triples:
(http://www.w3schools.com/rdf, author, 'Jan Egil refsnes')
(http://www.w3schools.com/rdf, homepage, http://www.w3schools.com)

Graph:



Document: 2 marks
Graph: 2 marks

b. *List out any three salient features of Big data.* *(3)*

Any three of the following feature to be listed
Volume: The quantity of generated and stored data.
Variety: The type and nature of the data.
Velocity: The speed at which the data is generated and processed.
Variability: Inconsistency of the data.
Veracity: The quality of captured data.

1 mark/feature = 3 marks

c. *How is GIS databases different from conventional databases?* *(3)*

Highlight any three of the following features with a one/two line explanation of the same.

(i) Location reference: Spatial location information has to be maintained in the form coordinates.
(ii) Temporal data: contains data that changes with respect to time.
(iii) Complex spatial information: spatial features such as points, lines, polygons etc. are to be represented and managed.
(iv) Field-based representation of the world as a continuous function of attributes that varies in spatially. **1 mark/feature = 3marks**

END