

184
14/11/23

184
Nov-23

Scoring Indicators

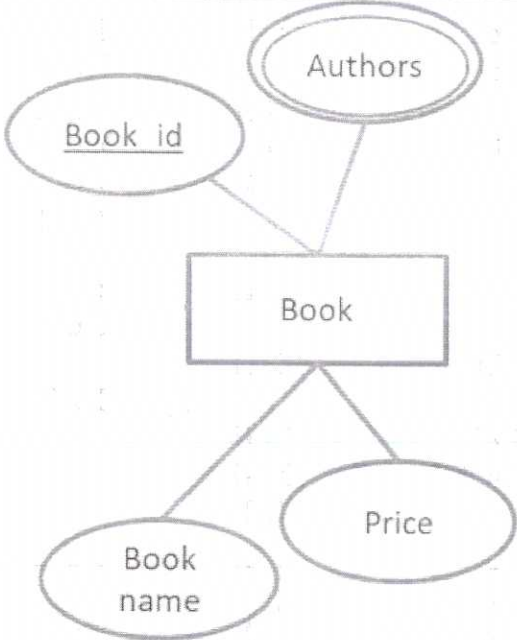
COURSE NAME : DATABASE MANAGEMENT SYSTEMS

COURSE CODE : 3133

QID : 2110220124

Q No	Scoring Indicators	Split score	Sub Total	Total score
PART A				9
I. 1	A software package/ system to facilitate the creation and maintenance of a computerized database.	1	1	
I. 2	Data Definition Language	1	1	
I. 3	SELECT * FROM STUDENT;	1	1	
I. 4	COUNT,SUM,AVG,MIN,MAX	2x0.5	1	
I. 5	Total participation, Partial participation	2x0.5	1	
I. 6	A domain is a collection of values that can be assigned to an attribute.	1	1	
I. 7	Strong Entity	1	1	
I. 8	Boyce-Codd Normal Form	1	1	
I. 9	The process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations	1	1	
PART B				24
II. 1	<ul style="list-style-type: none"> • <u>Self-describing nature of a database system:</u> A DBMS catalog stores the <i>description</i> of the database. The description is called meta-data). This allows the DBMS software to work with different databases. • <u>Insulation between programs and data:</u> Called program-data independence. Allows changing data storage structures and operations without having to change the DBMS access programs. • <u>Data Abstraction:</u> A data model is used to hide storage details and present the users with a <i>conceptual view</i> of the database. • <u>Support of multiple views of the data:</u> Each user may see a different view of the database, which describes <i>only</i> the data of interest to that user. • <u>Data Abstraction:</u> A data model is used to hide storage details and present the users with a <i>conceptual view</i> of the database. • <u>Support of multiple views of the data:</u> Each user may see a different view of the database, which 	Any 3x1	3	


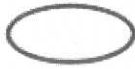






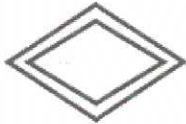
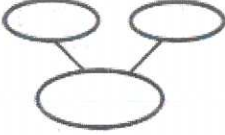

	describes <i>only</i> the data of interest to that user.			
II. 2	<p>When a schema at a lower level is changed, only the mappings between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence.</p> <ul style="list-style-type: none"> • Logical Data Independence: The capacity to change the conceptual schema without having to change the external schemas and their application programs. • Physical Data Independence: The capacity to change the internal schema without having to change the conceptual schema. 	1+1+1	3	
II. 3	<ul style="list-style-type: none"> • Superkey of R: A set of attributes SK of R such that no two tuples <i>in any valid relation instance</i> $r(R)$ will have the same value for SK. That is, for any distinct tuples t_1 and t_2 in $r(R)$, $t_1[SK] \neq t_2[SK]$. • Key of R: A "minimal" superkey; that is, a superkey K such that removal of any attribute from K results in a set of attributes that is not a superkey.. • If a relation has <i>several candidate keys</i>, one is chosen arbitrarily to be the primary key. The primary key attributes are <i>underlined</i>. 	1+1+1	3	
II. 4	<ul style="list-style-type: none"> • Centralized DBMS: combines everything into single system including- DBMS software, hardware, application programs and user interface processing software. <p style="text-align: center;">or</p> <p>The diagram illustrates a centralized DBMS architecture. At the top, 'Terminals' consist of multiple 'Display monitor' boxes connected to a central 'Network'. Below the network is the 'SOFTWARE' layer, which includes 'Application Programs', 'Terminal display control', 'Text editors', 'DBMS', and 'Compilers'. This layer sits on top of the 'Operating System'. The 'Operating System' layer contains a 'System bus' and 'Controllers'. Below the system bus is the 'HARDWARE/FIRMWARE' layer, which includes a 'CPU', 'Memory', 'Disk', and 'I/O devices (printers, tape drives ...)'. Arrows indicate the flow of data and control between these components.</p>	3	3	

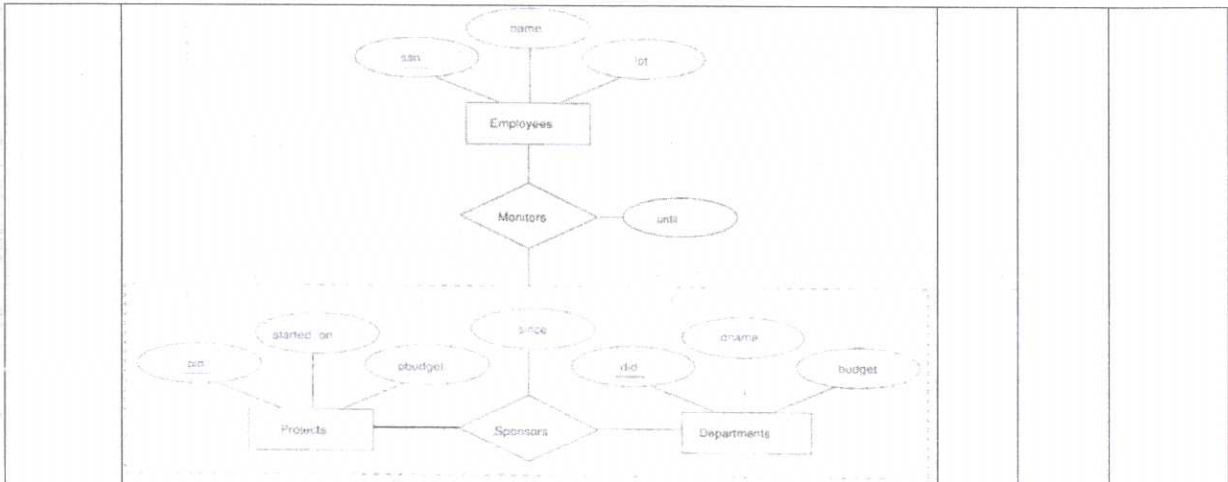
II. 5	<ul style="list-style-type: none"> SQL has a GROUP BY-clause for specifying the grouping attributes, which <i>must also appear in the SELECT-clause</i> The HAVING-clause is used for specifying a selection condition on groups (rather than on individual tuples) <u>Eg:</u> For each project <i>on which more than two employees work</i> , retrieve the project number, project name, and the number of employees who work on that project. <p>Q22:</p> <pre> SELECT PNUMBER, PNAME, COUNT (*) FROM PROJECT, WORKS_ON WHERE PNUMBER=PNO GROUP BY PNUMBER, PNAME HAVING COUNT (*) > 2; </pre>	3		
II. 6		3	3	
II. 7	<ul style="list-style-type: none"> ■ For each multivalued attribute A, create a new relation R. ■ This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute. ■ The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components. 	3	3	

II. 8	<p>Specialization</p> <ul style="list-style-type: none"> ● Is the process of defining a set of subclasses of a superclass ● The set of subclasses is based upon some distinguishing characteristics of the entities in the superclass ● Example: {SECRETARY, ENGINEER, TECHNICIAN} is a specialization of EMPLOYEE based upon <i>job type</i>. <ul style="list-style-type: none"> – May have several specializations of the same superclass <p>Generalization</p> <ul style="list-style-type: none"> ● The reverse of the specialization process ● Several classes with common features are generalized into a superclass; original classes become its subclasses ● Example: CAR, TRUCK generalized into VEHICLE; both CAR, TRUCK become subclasses of the superclass VEHICLE. 	3	3	
II.9	<ul style="list-style-type: none"> ● Simple <ul style="list-style-type: none"> – Each entity has a single atomic value for the attribute. For example, SSN or Sex. ● Composite <ul style="list-style-type: none"> – The attribute may be composed of several components. For example, Address (Apt#, House#, Street, City, State, ZipCode, Country) or Name (FirstName, MiddleName, LastName). Composition may form a hierarchy where some components are themselves composite. ● Multi-valued <ul style="list-style-type: none"> – An entity may have multiple values for that attribute. For example, Color of a CAR or PreviousDegrees of a STUDENT. Denoted as {Color} or {PreviousDegrees}. 	1+1+1	3	

	<p>Syntax for DML Commands</p> <p>The syntax for the DML commands are as follows –</p> <p>INSERT Insert command is used to insert data into a table.</p> <p>The syntax for insert command is as follows –</p> <p>Syntax Insert into <table_name> (column list) values (column values);</p> <p>SELECT Select command is used to retrieve data from the database.</p> <p>The syntax for the select command is as follows –</p> <p>Syntax SELECT * from <table_name>;</p> <p>DELETE Delete command is used to delete records from a database table.</p> <p>The syntax for the delete command is as follows -</p> <p>Syntax Delete from <table_name>WHERE condition;</p> <p>UPDATE Update command is used to update existing data within a table.</p> <p>The syntax for the update command is as follows -</p> <p>Syntax UPDATE <table_name> SET column_number =value_number WHERE condition;</p>			
	OR			
III. 4	<p>CREATE TABLE PROJECT(PNAME VARCHAR(20) NOT NULL, PNO INT PRIMARY KEY, PLOCATION VARCHAR(10)); INSERT VALUES INTO PROJECT('ABC',123,'XYZ'); UPDATE PROJECT SET PLOCATION='MUMBAI' WHERE PNO=3; RENAME TABLE PROJECT TO PROJECTS;</p>		7	7
III. 5	<ul style="list-style-type: none"> • A view is a “virtual” table that is derived from other tables • Allows for limited update operations (since the table may not physically be stored) 	4+3	7	7

	<ul style="list-style-type: none"> • Allows full query operations • A convenience for expressing certain operations • SQL command: CREATE VIEW <ul style="list-style-type: none"> • a table (view) name • a possible list of attribute names (for example, when arithmetic operations are specified or when we want the names to be different from the attributes in the base relations) • a query to specify the table contents • Specify a different WORKS_ON table CREATE VIEW WORKS_ON_NEW AS SELECT FNAME, LNAME, PNAME, HOURS FROM EMPLOYEE, PROJECT, WORKS_ON WHERE SSN=ESSN AND PNO=PNUMBER GROUP BY PNAME; 			
	OR			
III. 6	a) SELECT * FROM STUDENT WHERE AGE>20; b) SELECT * FROM STUDENT WHERE NAME LIKE 'S%'; c) SELECT SUM(MARK) FROM STUDENT; d) SELECT * FROM STUDENT WHERE MARK=SELECT MAX(MARK) FROM STUDENT;		7	7

III. 7		Represents Entity	ANY 7X1	7	7
		Represents Attribute			
		Represents Relationship			
		Links Attribute(s) to entity set(s) or Entity set(s) to Relationship set(s)			
		Represents Multivalued Attributes			
		Represents Derived Attributes			
		Represents Total Participation of Entity			
		Represents Weak Entity			
		Represents Weak Relationships			
		Represents Composite Attributes			
		Represents Key Attributes / Single Valued Attributes			
	OR				
III. 8	<ul style="list-style-type: none"> ● Used when we have to model a relationship involving (entity sets and) a <i>relationship set</i>. <ul style="list-style-type: none"> – <i>Aggregation</i> allows us to treat a relationship set as an entity set for purposes of participation in (other) relationships. 		3+4	7	7



<p>III. 9</p>	<p>Functional Dependencies</p> <ul style="list-style-type: none"> ➤ Functional dependencies (FDs) are used to specify <i>formal measures</i> of the "goodness" of relational designs ➤ FDs and keys are used to define normal forms for relations ➤ FDs are constraints that are derived from the <i>meaning</i> and <i>interrelationships</i> of the data attributes ➤ A set of attributes X <i>functionally determines</i> a set of attributes Y if the value of X determines a unique value for Y ➤ $X \rightarrow Y$ holds if whenever two tuples have the same value for X, they <i>must have</i> the same value for Y ➤ For any two tuples t1 and t2 in any relation instance r(R): <i>If</i> $t1[X]=t2[X]$, <i>then</i> $t1[Y]=t2[Y]$ ➤ $X \rightarrow Y$ in R specifies a <i>constraint</i> on all relation instances r(R) ➤ Written as $X \rightarrow Y$; can be displayed graphically on a relation schema as in Figures. (denoted by the arrow:). ➤ FDs are derived from the real-world constraints on the attributes <p>Examples of FD constraints</p> <ul style="list-style-type: none"> ➤ social security number determines employee name SSN \rightarrow ENAME ➤ project number determines project name and location PNUMBER \rightarrow {PNAME, PLOCATION} ➤ employee ssn and project number determines the hours per week that the employee works on the project {SSN, PNUMBER} \rightarrow HOURS 	<p>4+3</p>	<p>7</p>	<p>7</p>
---------------	--	------------	----------	----------

	<ul style="list-style-type: none"> ➤ An FD is a property of the attributes in the schema R ➤ The constraint must hold on <i>every relation instance</i> $r(R)$ ➤ If K is a key of R, then K functionally determines all attributes in R (since we never have two distinct tuples with $t1[K]=t2[K]$) 			
	OR			
III. 10	<p>Why Concurrency Control is needed:</p> <ul style="list-style-type: none"> ● The Lost Update Problem. This occurs when two transactions that access the same database items have their operations interleaved in a way that makes the value of some database item incorrect. ● The Temporary Update (or Dirty Read) Problem. This occurs when one transaction updates a database item and then the transaction fails for some reason. The updated item is accessed by another transaction before it is changed back to its original value. ● The Incorrect Summary Problem . If one transaction is calculating an aggregate summary function on a number of records while other transactions are updating some of these records, the aggregate function may <u>calculate some values before they are updated and others after they are updated.</u> 	2+2+3	7	7
III. 11	<ul style="list-style-type: none"> ● A relation schema R is in 4NF with respect to a set of dependencies F (that includes functional dependencies and multivalued dependencies) if, for every <i>nontrivial</i> multivalued dependency $X \twoheadrightarrow Y$ in F^+, X is a superkey for R. <p>Note: F^+ is the (complete) set of all dependencies (functional or multivalued) that will hold in every relation state r of R that satisfies F. It is also called the closure of F.</p> <ul style="list-style-type: none"> ● Decomposing a relation state of EMP that is not in 4NF. (a) EMP relation with additional tuples. (b) Two corresponding 4NF relations EMP_PROJECTS and EMP_DEPENDENTS. 	3+4	7	7

	<p>(a) EMP</p> <table border="1" data-bbox="414 257 901 336"> <thead> <tr> <th>ENAME</th> <th>PNAME</th> <th>DNAME</th> </tr> </thead> <tbody> <tr><td>Smith</td><td>X</td><td>John</td></tr> <tr><td>Smith</td><td>Y</td><td>Anna</td></tr> <tr><td>Smith</td><td>X</td><td>Anna</td></tr> <tr><td>Smith</td><td>Y</td><td>John</td></tr> <tr><td>Brown</td><td>W</td><td>Jim</td></tr> <tr><td>Brown</td><td>X</td><td>Jim</td></tr> <tr><td>Brown</td><td>Y</td><td>Jim</td></tr> <tr><td>Brown</td><td>Z</td><td>Jim</td></tr> <tr><td>Brown</td><td>W</td><td>Joan</td></tr> <tr><td>Brown</td><td>X</td><td>Joan</td></tr> <tr><td>Brown</td><td>Y</td><td>Joan</td></tr> <tr><td>Brown</td><td>Z</td><td>Joan</td></tr> <tr><td>Brown</td><td>W</td><td>Bob</td></tr> <tr><td>Brown</td><td>X</td><td>Bob</td></tr> <tr><td>Brown</td><td>Y</td><td>Bob</td></tr> <tr><td>Brown</td><td>Z</td><td>Bob</td></tr> </tbody> </table>	ENAME	PNAME	DNAME	Smith	X	John	Smith	Y	Anna	Smith	X	Anna	Smith	Y	John	Brown	W	Jim	Brown	X	Jim	Brown	Y	Jim	Brown	Z	Jim	Brown	W	Joan	Brown	X	Joan	Brown	Y	Joan	Brown	Z	Joan	Brown	W	Bob	Brown	X	Bob	Brown	Y	Bob	Brown	Z	Bob			
ENAME	PNAME	DNAME																																																					
Smith	X	John																																																					
Smith	Y	Anna																																																					
Smith	X	Anna																																																					
Smith	Y	John																																																					
Brown	W	Jim																																																					
Brown	X	Jim																																																					
Brown	Y	Jim																																																					
Brown	Z	Jim																																																					
Brown	W	Joan																																																					
Brown	X	Joan																																																					
Brown	Y	Joan																																																					
Brown	Z	Joan																																																					
Brown	W	Bob																																																					
Brown	X	Bob																																																					
Brown	Y	Bob																																																					
Brown	Z	Bob																																																					
	OR																																																						
<p>III. 12</p>	<p><u>Desirable Properties of Transactions</u></p> <p>ACID properties:</p> <ul style="list-style-type: none"> ● Atomicity: A transaction is an atomic unit of processing; it is either performed in its entirety or not performed at all. ● Consistency preservation: A correct execution of the transaction must take the database from one consistent state to another. ● Isolation: A transaction should not make its updates visible to other transactions until it is committed; this property, when enforced strictly, solves the temporary update problem and makes cascading rollbacks of transactions unnecessary . ● Durability or permanency: Once a transaction changes the database and the changes are committed, these changes must never be lost because of subsequent failure. 	<p>Any 3 Listin g-1 Expln -6</p>	7	7																																																			