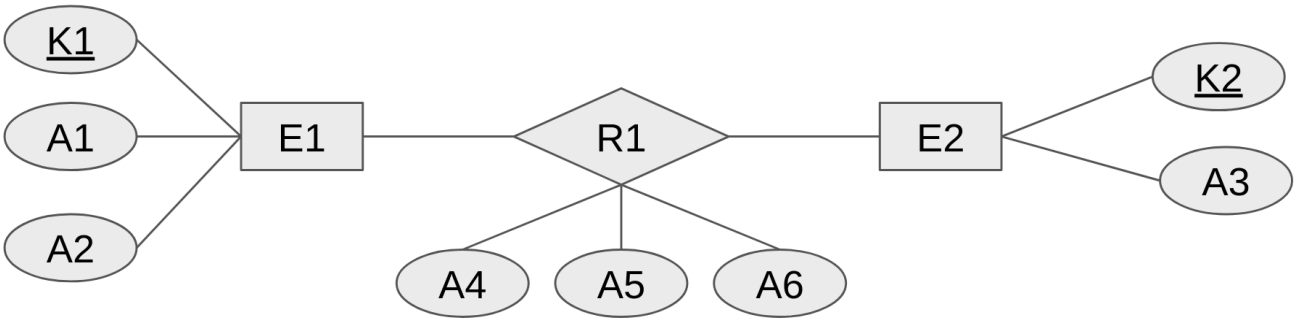
Started on	Tuesday, 8 April 2025, 09:23
State	Finished
Completed on	Tuesday, 8 April 2025, 09:24
Time taken	1 min 7 secs
Marks	0.00/124.00
Grade	**0.00** out of 10.00 (**0**%)

Question 1

Not answered

Marked out of 4.00



The inventory system of a Switch game card rental store tracks the information of customers, cards, and rental transactions between customers and cards. Customers are identified with their names, telephone numbers, and unique IDs, while each game card is associated with a short introduction and one unique ID. To better track the availability of game cards, the store needs to record the borrow time, agreed return time, and final return time of each transaction.

Below is an ER model to describe the above scenario.
Fill in the gaps with the following words. Words can be reused multiple times or not be used at all.

Note: If multiple words can be filled in gaps without distinction, please place them in alphabetical order.

Entity E1 : Key K1 :

Attribute A1 : Attribute A2 :

Entity E2 : Key K2 :

Attribute A3 : Relationship R1:

Attribute A4 : Attribute A5 :

Attribute A6 :

Introduction	Customer name	Agreed return time	Customer ID
Card ID	Game cards	Return time	Customer telephone number
Customers	Borrow time	Borrow	

Question 2

Not answered

Marked out of 2.00

Which of the following questions can be answered from the ER diagram in Question 1?

- a. Which game cards are the most popuplar (borrowed the most)?
- b. Who are the customers that borrowed the most?
- c. Who are the customers that did not return cards on time?
- d. Who will be the next customers to borrow one specific card?

Question 3

Not answered

Marked out of 2.00

What is the type of relationship R1 between the entity E1 and E2 in Question 1?

- a. One-to-One
- b. Many-to-one
- c. Many-to-Many
- d. One-to-Many

Question 4

Not answered

Marked out of 2.00

Consider the ER model introduced in Question 1, and fill in the gaps in the sentences.

Words can be reused multiple times or not be used at all.

The participation of the E1 entity in the R1 relationship is [ ]

The participation of the E2 entity in the R1 relationship is [ ]

Partial | Total

Question 5

Not answered

Marked out of 2.00

Are E1 and E2 in Question 1 weak entities?

- a. E2 is a weak entity but E1 is not a weak entity
- b. None of them is a weak entity
- c. E1 is a weak entity but E2 is not a weak entity
- d. Both are weak entities

Question 6

Not answered

Marked out of 2.00

Consider a weak entity set W and its identifying entity set O. The primary key of W is composed of

a. All attributes of W and all attributes of O.

b. The partial key of W and the primary key of O.

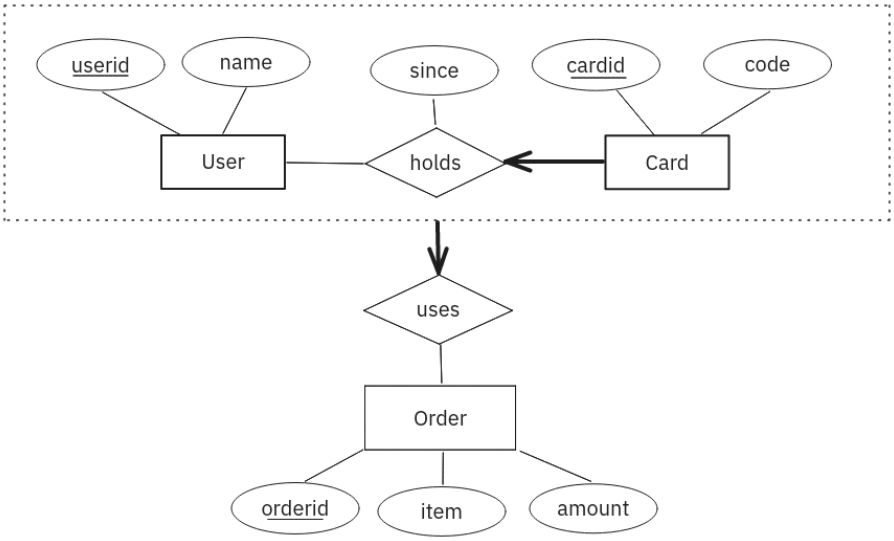
c. All attributes of W and the primary key of O.

d. The partial key of W and all attributes of O.

Information

Suppose a fast food company hands out membership cards to their users, which they can choose to enter when ordering food. It is however not required to be a user, nor to have a card, in order to make an order.

Consider the following translation from an ER model to the relational model



And the following relational model:

- `User(userid, name)`
- `Card(cardid, code)`
- `UserHolds(userid, cardid, since)`
- `Order(orderid, item, amount)`
- `OrderUses(userid, cardid, orderid)`

Where fields underlined are the primary keys of the relation. All fields are not null.

Question 7

Not answered

Marked out of 2.00

Choose the **fewest** number of choices of key constraints to be added, so the relational model completely represents all key constraints of the ER model.

- a. Foreign keys: OrderUses(userid) REFERENCES User(userid), OrderUses(cardid) REFERENCES Card(cardid)
- b. Unique: OrderUses(userid, cardid)
- c. Unique: UserHolds(cardid)
- d. Unique: UserHolds(userid)
- e. Unique: OrderUses(orderid)
- f. Foreign keys: OrderUses(orderid) REFERENCES Order(orderid)
- g. Foreign keys: OrderUses(userid, cardid) REFERENCES UserHolds(userid, cardid)
- h. Foreign keys: UserHolds(userid) REFERENCES User(userid), UserHolds(cardid) REFERENCES Card(cardid)

Question 8

Not answered

Marked out of 1.00

Is it possible to merge the tables Order and OrderUses?

- a. Yes
- b. No

Question 9

Not answered

Marked out of 1.00

Is it possible to merge UserHolds with another table?

- a. Yes, UserHolds can be merged with User
- b. Yes, UserHolds can be merged with Card
- c. Yes, UserHolds can be merged with OrderUses
- d. No, UserHolds can not be merged at all

Question 10

Not answered

Marked out of 1.00

Suppose we want to enforce that each card can only be used on one order. Which constraints have to be added into OrderUses?

a. UNIQUE (cardid)

b. UNIQUE (orderid)

c. UNIQUE (cardid, orderid)

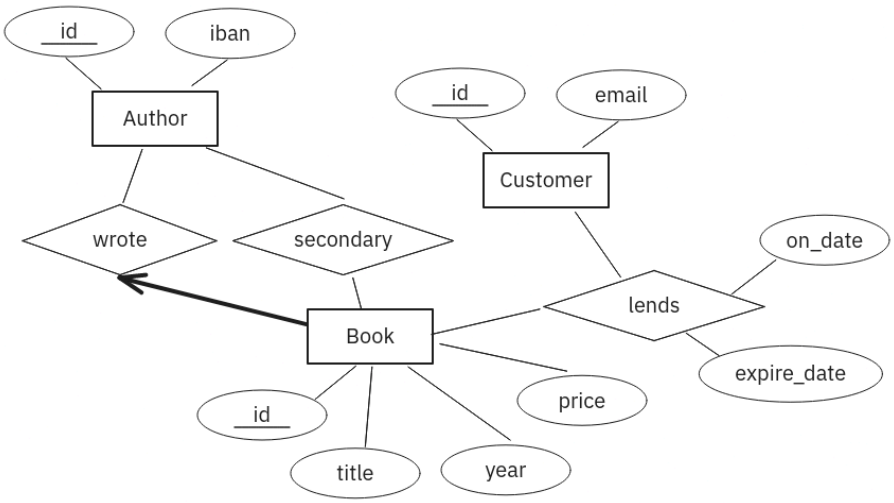
d. No constraints needs to be added, it is already enforced with the current constraints

Information

Suppose a book publishing company wants to digitally lend its books to customers and transfer parts of its lending price to the books' authors. The company wants to store the following:

- For each author, their id and IBAN account number.
- For each book, its title, year, one primary author, zero or more secondary authors and its lending price.
- For each customer, their email address, which should be unique to the user.
- A record of all current lendings, each consisting of a customer lending a single book with an expiration date, and the date the transaction was made.
- Finally, as a measure against mass lending, each customer can only lend **one book per day**.

The company came up with the following ER model:



Question 11

Not answered

Marked out of 10.00

Finish the following SQL statements to create the relational model corresponding to the requirements, guided by the ER model.

Assume that all string fields are at most 100 characters long, and we only use CHAR(..) (where .. is filled with a number), INTEGER, DATE and FLOAT for data types.

If multiple columns are needed, list them in the order they are defined in the table, with one comma and one space in between (for example: “first_col, second_col”).

```
CREATE TABLE Author (
    id INTEGER NOT NULL,
    iban CHAR(25) NOT NULL UNIQUE,
    PRIMARY KEY (id)
);
CREATE TABLE Book (
    id INTEGER NOT NULL,
    title [ ] NOT NULL,
    year [ ] NOT NULL,
    price FLOAT NOT NULL,
    primary_author [ ] NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (primary_author) REFERENCES [ ] ( [ ] )
);
CREATE TABLE SecondaryAuthor (
    author_id [ ] NOT NULL,
    book_id [ ] NOT NULL,
    PRIMARY KEY ( [ ] ),
    FOREIGN KEY (author_id) REFERENCES [ ] ( [ ] ),
    FOREIGN KEY (book_id) REFERENCES [ ] ( [ ] )
);
CREATE TABLE Customer (
    id CHAR(100) NOT NULL,
    email [ ] NOT NULL,
    UNIQUE( [ ] ),
    PRIMARY KEY (id)
);
CREATE TABLE Lend(
    customer_id [ ] NOT NULL,
    book_id [ ] NOT NULL,
    on_date DATE NOT NULL,
    expiry_date DATE NOT NULL,
    PRIMARY KEY ( [ ] ),
    UNIQUE ( [ ] ),
    FOREIGN KEY (customer_id) REFERENCES Customer(id),
    FOREIGN KEY (book_id) REFERENCES [ ] ( [ ] )
);
```

Question 12

Not answered

Marked out of 2.00

The following is the description of the database. The questions come afterwards.

Pubs Database Schema
author(author_id, f irst name, last name)
author_pub(author_id, pub_id, author position)
book(book_id, book_title, month, year, editor)
pub(pub_id, title, book_id)
• author_id in author_pub is a foreign key referencing author
• pub_id in author_pub is a foreign key referencing pub
• book_id in pub is a foreign key referencing book
• editor in book is a foreign key referencing author(author_id)
• Primary keys are underlined

Pubs Database State:

$r(author)$

author_id	first_name	last_name
1	John	McCarthy
2	Dennis	Ritchie
3	Ken	Thompson
4	Claude	Shannon
5	Alan	Turing
6	Alonzo	Church
7	Perry	White
8	Moshe	Vardi
9	Roy	Batty

$r(author\_pub)$

author_id	pub_id	author_position
1	1	1
2	2	1
3	2	2
4	3	1
5	4	1
5	5	1
6	6	1

$r(book)$

book_id	book_title	month	year	editor
1	CACM	April	1960	8
2	CACM	July	1974	8
3	BST	July	1948	2
4	LMS	November	1936	7
5	Mind	October	1950	NULL
6	AMS	Month	1941	NULL
7	AAAI	July	2012	9
8	NIPS	July	2012	9

$r(pub)$

pub_id	title	book_id
1	LISP	1
2	Unix	2
3	Info Theory	3
4	Turing Machines	4
5	Turing Test	5
6	Lambda Calculus	6

How many tuples will be returned by the following relational algebra query?
$\pi_{book\_title}(book)$

- a. 6
- b. 9
- c. 8
- d. 7

Question 13

Not answered

Marked out of 2.00

How many tuples are returned by the following relational algebra expression?

$|\pi_{author\_id}(author) - \pi_{editor}(book)|$

a. 9

b. 5

c. 8

d. 10

e. 0

f. 1

Question 14

Not answered

Marked out of 3.00

Which of the following relational algebra expressions returns the names of all authors who are book editors?

a. $\pi_{first\_name,last\_name}(author \bowtie_{author\_id=author\_id} author\_pub)$

b. $\pi_{first\_name,last\_name}(author \bowtie_{author\_id=editor} book)$

c. $\pi_{first\_name,last\_name}(author \bowtie_{author\_id=pub\_id} author\_pub)$

d. $\pi_{first\_name,last\_name}(author \bowtie_{author\_id=book\_id} pub)$

Question 15

Not answered

Marked out of 3.00

Which of the followings is a relational algebra expression that returns the names of all authors who are not book editors?

a. $\pi_{first\_name,last\_name}(author) - \pi_{first\_name,last\_name}(author \bowtie_{author\_id=editor} book)$

b. $\pi_{first\_name,last\_name}(author) \cup \pi_{first\_name,last\_name}(author \bowtie_{author\_id=editor} book)$

c. $\pi_{first\_name,last\_name}((\pi_{author\_id}(author) - \pi_{editor}(book)) * author)$

d. $\pi_{first\_name,last\_name}(author) \cap \pi_{first\_name,last\_name}(author \bowtie_{author\_id=editor} book)$

Question 16

Not answered

Marked out of 3.00

What question does the following relational algebra expression answer?

$author * (author\_pub * (\sigma_{month='July'}(book) * pub))$

a. Which authors authored a pub that was published in July?

b. Which authors edited books that were published in July?

c. Which authors were born in July?

Question 17

Not answered

Marked out of 2.00

Which SQL statement is used to retrieve unique values from a column?

a. SELECT column_name FROM table_name;

b. SELECT DISTINCT column_name FROM table_name;

c. SELECT UNIQUE column_name FROM table_name;

d. SELECT DIFFERENT column_name FROM table_name;

Question 18

Not answered

Marked out of 2.00

What does the SQL function COUNT(*) do?

a. Counts all non-null values in a column.

b. Counts all rows in a table, including those with null values

c. Counts all columns in a table.

d. Counts all distinct values in a column.

Question 19

Not answered

Marked out of 2.00

How do you select all columns from a table named ‘employees’?

a. SELECT all FROM employees;

b. SELECT columns FROM employees;

c. SELECT employees.*;

d. SELECT * FROM employees;

Question 20

Not answered

Marked out of 1.00

Which SQL statement is used to remove a table from a database?

a. TRUNCATE TABLE table_name;

b. DROP TABLE table_name;

c. REMOVE TABLE table_name;

d. DELETE TABLE table_name;

Question 21

Not answered

Marked out of 2.00

What is the RAID level that provides exact mirroring without handling corruption?

a. RAID 0

b. RAID 1

c. RAID 5

d. RAID 3

Question 22

Not answered

Marked out of 2.00

Which of these are the responsibilities of the buffer manager.

a. Reduces IO by buffering/caching

b. Uses logging that can be used for recovery

c. Keeps active pages in memory

d. Discards/write back pages when needed

Information

You have designed a buffer pool capable of storing up to 4 frames at a given time.

Assuming you have three corresponding access traces (one for each policy) where each letter represents a requested page ID. The trace starts from the left and goes right in the discrete time.

While answering the questions, assume you start from an empty buffer pool for each policy, and you don't need to consider the pin-count for pages, and no page is written on.

Information

(A) For below two questions (A.1, A.2) use the following trace:

1 2 5 4 1 6 3 1 3 5 1 7 2

Question 23

Not answered

Marked out of 2.00

(A.1) If you use **FIFO** algorithm for page replacement, how many I/O does the buffer manager need to issue?

Answer: ✕

Question 24

Not answered

Marked out of 2.00

(A.2) If you use **FIFO** algorithm for page replacement, which pages will remain in the buffer pool at the end of trace?

- a. [2] [3] [5] [7]
- b. [2] [3] [5] [6]
- c. [1] [2] [6] [7]
- d. [1] [2] [5] [7]
- e. [1] [3] [5] [6]

Information

(B) For below two questions (B.1, B.2) use the following trace:

2 1 4 5 1 6 3 1 3 5 1 7 2

Question 25

Not answered

Marked out of 2.00

(B.1) If you use **LRU** algorithm for page replacement, how many I/O does the buffer manager need to issue?

Answer: ✕

Question 26

Not answered

Marked out of 2.00

(B.2) If you use **LRU** algorithm for page replacement, which pages will remain in the buffer pool at the end of trace?

- a. [1] [2] [5] [7]
- b. [1] [2] [3] [7]
- c. [1] [3] [5] [7]
- d. [1] [4] [6] [7]
- e. [1] [2] [3] [5]

Information

(C) For below two questions (C.1, C.2) use the following trace:

5 6 2 4 1 3 6 1 3 5 1 2 7

Question 27

Not answered

Marked out of 2.00

(C.1) If you use **CLOCK** algorithm for page replacement, how many I/O does the buffer manager need to issue?

Answer: ✕

Question 28

Not answered

Marked out of 2.00

(C.2) If you use **CLOCK** algorithm for page replacement, which pages will remain in the buffer pool at the end of trace?

- a. [1] [2] [6] [7]
- b. [2] [3] [5] [7]
- c. [1] [3] [5] [7]
- d. [2] [5] [6] [7]
- e. [1] [2] [3] [7]
- f. [1] [2] [5] [7]

Question 29

Not answered

Not graded

(Optional) You can briefly explain your answers to the questions above. You can use free text or formulas to explain.

Question 30

Not answered

Marked out of 2.00

DataFlex Inc. stores its customer records in a database with the following schema: CUSTOMER(custID, firstName, lastName, address, city, country, joinDate, lastOrder)

The size of each data record is 256B with the following attribute sizes: custID is 8B, firstName is 32B, lastName is 32B, address is 64B, city is 32B, country is 16B, joinDate is 8B, and lastOrder is 64B (total of 256B per record).

The disk page size is 4096B. Each page can store only a whole number of records, and every page contains a 32B header and 32B footer of metadata.

How many records fit on a single page?

Answer: ✕

Question 31

Not answered

Marked out of 1.00

Using the same CUSTOMER table and page information from the previous question, what is the size of the CUSTOMER file in number of disk pages if it contains 50,000 customer records?

Answer: ✕

Question 32

Not answered

Marked out of 2.00

[Record Storage] Answer the following questions (marked with [Record Storage]) assuming the following scenario.

Consider the TravelAgency database that uses the PAX (Partition Attributes Across) storage model for its BOOKING table: BOOKING(bookingID, customerName, destination, departureDate, returnDate, price)

The size breakdown per attribute is as follows: bookingID is 8B, customerName is 48B, destination is 32B, departureDate is 8B, returnDate is 8B, and price is 8B (total of 112B per record).

The disk page size is 2048B: each page contains a 32B header and no footer. Each minipage within a PAX page also has a 4B header.

[Record Storage] How many records fit in a single PAX page?

Answer: ✕

Question 33

Not answered

Marked out of 3.00

[Record Storage] Calculate the offset (in bytes) from the beginning of the page to the start of the price minipage **records** in the PAX layout.

- a. 1960
- b. 1800
- c. 1824
- d. 1768

Question 34

Not answered

Marked out of 1.00

In NSM (N-ary Storage Model), if a query needs to access only one attribute from a table, it needs to read all attributes of all records.

- True
- False

Question 35

Not answered

Marked out of 1.00

DSM (Decomposition Storage Model) typically has better compression rates than NSM because values of the same attribute are stored together.

- True
- False

Question 36

Not answered

Marked out of 1.00

PAX (Partition Attributes Across) storage model is optimized for analytic processing workloads but performs poorly for transaction processing workloads compared to NSM.

- True
- False

Question 37

Not answered

Marked out of 1.00

[I/O Cost] Answer the following questions (marked with [I/O Cost]) assuming the following scenario.

InvestBank maintains a TRANSACTION table to record all financial transactions: TRANSACTION(transID, accountID, timestamp, amount, type, description)

The file contains 1 million records and is stored as a heap file with 20 records per page (50,000 pages total). The database uses NSM storage model, and the file is not sorted on any attribute.

[I/O Cost] The I/O cost (number of page accesses) for finding all transactions for accountID = 54321 without any index is:

Answer: ✕

Question 38

Not answered

Marked out of 1.00

[I/O Cost] The TRANSACTION file now has an unclustered B+ tree index on the accountID attribute. Each account has exactly 100 transactions on average. The I/O cost (number of page accesses) to find all transactions for accountID = 54321 is: (Assume the B+ tree has a height of 3 and the index entries are much smaller than data records)

Answer: ✕

Question 39

Not answered

Marked out of 2.00

For queries that need to retrieve records in a specific range of values (e.g., all transactions with amount between $1,000 and $5,000), the most efficient file organization is:

a. Hash index

b. B+ tree index

c. Heap file

d. Sorted file

Question 40

Not answered

Marked out of 1.00

You are designing a database that requires fast insertions and point lookups but does not need range queries. Which file organization should you choose?

a. Sorted file

b. Heap file

c. Hash index

d. B+ tree index

Question 41

Not answered

Marked out of 3.00

Given the employee dataset:

EMPLOYEES(jobid, name, age, department, ranking)

The size of each record is 120B: jobid (primary key) is 10B, name is 50B, age is 4B, department is 44B, ranking is 12B. The records are organized in 8KB pages (each page stores an integer number of records, and you can ignore metadata). You can assume that the distribution of rankings follows a Gaussian curve from 1-10, the age distribution is typical of a mid-sized tech company workforce, and the most popular department is "Engineering" while the least popular department is "Legal".

Which index to use for that query, depending on the following query patterns and dataset sizes? If it would be better to use no index than the possible options, leave each dropdown as No Index.

(1)

Dataset size of 10.5 million records with infrequent modifications and frequent queries of the format SELECT * FROM employees WHERE age < 21

Data structure is:

Question 42

Not answered

Marked out of 3.00

Format is

Question 43

Not answered

Marked out of 5.00

On what key:

Question 44

Not answered

Marked out of 3.00

Density is

Question 45

Not answered

Marked out of 3.00

Dataset size of 500,000 records with frequent updates and queries of the format SELECT jobid FROM employees WHERE department="Legal"

Data structure is:

Question 46

Not answered

Marked out of 3.00

Format is:

Question 47

Not answered

Marked out of 5.00

On what key:

Question 48

Not answered

Marked out of 3.00

Density is:

Question 49

Not answered

Marked out of 3.00

Dataset size of 40 records with frequent updates and queries of the format SELECT AVG(ranking) FROM employees WHERE age > 23 AND department="Engineering"

Data structure is:

Question 50

Not answered

Marked out of 3.00

Format is:

Question 51

Not answered

Marked out of 6.00

On what key:

Question 52

Not answered

Marked out of 3.00

Density is:

◄ MIDTERM Attendance

Jump to...

Lecture (hidden) ►