Cambridge International AS & A Level

COMPUTER SCIENCE **9618/12**

Paper 1 Theory Fundamentals **May/June 2021**

1 Raj owns houses that other people rent from him. He has a database that stores details about the people who rent houses, and the houses they rent. The database, HOUSE_RENTALS, has the following structure:

```
CUSTOMER(CustomerID, FirstName, LastName, DateOfBirth, Email)
HOUSE(HouseID, HouseNumber, Road, Town, Bedrooms, Bathrooms)
RENTAL(RentalID, CustomerID, HouseID, MonthlyCost, DepositPaid)
```

(a) Give the definition of the following database terms, using an example from the database HOUSE_RENTALS for each definition.

Term	Definition and example
Field	
Entity	
Foreign key	

(b) Tick (✓) **one** box to identify whether the database HOUSE_RENTALS is in Third Normal Form (3NF) or not in 3NF.
Justify your choice using one or more examples from the database HOUSE_RENTALS.

In 3NF	
Not in 3NF	

Justification .....

(c) Example data from the table RENTAL are given:

RentalID	CustomerID	HouseID	MonthlyCost	DepositPaid
1	22	15B5L	1000.00	Yes
2	13	3F	687.00	No
3	1	12AB	550.00	Yes
4	3	37	444.50	Yes

(i) Complete the following Data Definition Language (DDL) statement to define the table RENTAL.

```
CREATE ..... ..... (
      RentalID INTEGER NOT NULL,
      CustomerID INTEGER NOT NULL,
      HouseID ..... (5) NOT NULL,
      MonthlyCost ..... NOT NULL,
      DepositPaid BOOLEAN NOT NULL,
      ..... (RentalID)
);
```

(ii) Write a Data Manipulation Language (DML) script to return the first name and last name of all customers who have not paid their deposit.

.....

Answer:

(a)

<table>
<tr><th>Term</th><th>Definition and example</th></tr>
<tr><td>Field</td><td>A field is a single attribute (piece of information) stored about an entity in a database table. Each field becomes a column in the table.

Example from HOUSE_RENTALS:
In the CUSTOMER table:

<table><tr><th>CustomerID</th><th>FirstName</th><th>LastName</th><th>DateOfBirth</th><th>Email</th></tr></table></td></tr>
<tr><td>Entity</td><td>An entity is a real-world object, person, place, or thing about which the database stores information. Each entity is represented by a table.
Examples from HOUSE_RENTALS:
The database contains three entities:
CUSTOMER
HOUSE
RENTAL</td></tr>
<tr><td>Foreign key</td><td>A foreign key is a field in one table that refers to the primary key of another table. It is used to create a relationship between the tables.
Example from HOUSE_RENTALS:

CUSTOMER Table

<table><tr><th>CustomerID (Primary Key)</th><th>FirstName</th></tr><tr><td>C001</td><td>Ali</td></tr><tr><td>C002</td><td>Sara</td></tr></table>
RENTAL Table<table><tr><th>RentalID</th><th>CustomerID</th><th>HouseID</th><th>MonthlyCost</th></tr><tr><td>R101</td><td>C001</td><td>H201</td><td>1200</td></tr></table>
Here, CustomerID in the RENTAL table is a foreign key because it refers to CustomerID (the primary key) in the CUSTOMER table.</td></tr>
</table>

(b)

☑ In 3NF

☐ Not in 3NF

Justification

The database is **in Third Normal Form (3NF)** because:

1. **Each table has a primary key** that uniquely identifies each record.
   - **CUSTOMER** → `CustomerID`
   - **HOUSE** → `HouseID`
   - **RENTAL** → `RentalID`
2. **Each table stores information about only one entity.**
   - The **CUSTOMER** table stores only customer details.
   - The **HOUSE** table stores only house details.
   - The **RENTAL** table stores only rental information.
3. **There are no repeating groups or duplicate data** within the tables.
4. **Non-key attributes depend only on the primary key**, not on another non-key attribute.
   - In the **CUSTOMER** table, `FirstName`, `LastName`, `DateOfBirth`, and `Email` all depend only on `CustomerID`.
   - In the **HOUSE** table, `HouseNumber`, `Road`, `Town`, `Bedrooms`, and `Bathrooms` all depend only on `HouseID`.
   - In the **RENTAL** table, `MonthlyCost` and `DepositPaid` depend only on `RentalID`.
5. **Relationships between tables are created using foreign keys**, avoiding data duplication.
   - `CustomerID` in the **RENTAL** table is a foreign key linking to the **CUSTOMER** table.
   - `HouseID` in the **RENTAL** table is a foreign key linking to the **HOUSE** table.

(C) (i)

```
CREATE TABLE RENTAL (
    RentalID INTEGER NOT NULL,
    CustomerID INTEGER NOT NULL,
    HouseID VARCHAR(5) NOT NULL,
    MonthlyCost DECIMAL(6,2) NOT NULL,
    DepositPaid BOOLEAN NOT NULL,
    PRIMARY KEY (RentalID)
);
```

Explanation of the Missing Parts

Missing Part	Answer	Reason
TABLE	TABLE	Used to create a new table.
RENTAL	RENTAL	Name of the table being created.
VARCHAR(5)	VARCHAR(5)	HouseID contains letters and numbers (e.g., 15B5L, 12AB), so it is a text field with a maximum length of 5 characters.
DECIMAL(6,2)	DECIMAL(6,2)	MonthlyCost stores monetary values such as 1000.00 and 444.50.
PRIMARY KEY	PRIMARY KEY	Identifies RentalID as the unique identifier for each record.

(c)(ii) Data Manipulation Language (DML)

Question: Return the **first name** and **last name** of all customers who **have not paid their deposit**.

Since:

- FirstName and LastName are stored in the **CUSTOMER** table.
- DepositPaid is stored in the **RENTAL** table.

We must **join** the two tables using CustomerID.

SQL Query

```
SELECT FirstName, LastName
FROM CUSTOMER, RENTAL
WHERE CUSTOMER.CustomerID = RENTAL.CustomerID
AND DepositPaid = No;
```

Alternative SQL (using JOIN syntax)

```
SELECT FirstName, LastName
FROM CUSTOMER
INNER JOIN RENTAL
ON CUSTOMER.CustomerID = RENTAL.CustomerID
WHERE DepositPaid = No;
```

Both queries produce the same result.