# Introduction to databases

| | |
|---|---|
| **Iniziato** | domenica, 4 luglio 2021, 13:29 |
| **Stato** | Completato |
| **Terminato** | domenica, 4 luglio 2021, 13:36 |
| **Tempo impiegato** | 6 min. 42 secondi |
| **Valutazione** | Non ancora valutato |

**Domanda 1**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

The primary key of a table

_____

- (a) must be referenced by a foreign key
- (b) may not be composed of a single element
- (c) must be unique but it might not be minimal
- (d) may not be composite
- (e) none of the answers are correct ✓

La risposta corretta è: none of the answers are correct

**Domanda 2**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

The HTML statement:

```
<select name = "course">
   <option value = "23ACIPL"> Mathematical analysis II </option>
   <option value = "08CKRPL"> Statistics </option>
   <option value = "14AFQPL" selected> Databases </option>
</ Select>
```

- (a) None of the other answers is correct ✓
- (b) Creates a set of checkboxes
- (c) It isn't correct because the value attribute only accepts integers
- (d) Creates a set of radio buttons
- (e) Creates a drop-down list where the default value is Mathematical Analysis II

La risposta corretta è: None of the other answers is correct

**Domanda 3**

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

A transaction has the property of durability if

- (a) all of the operations composing it are either completed, or they are undone, as if they had never been executed
- (b) it makes modifications permanent immediately after the transaction has ended ✓
- (c) it is executed on the system at the same time as other transactions as if it were the only one being executed
- (d) it takes the system from a valid state to another valid state
- (e) none of the answers are correct

La risposta corretta è: it makes modifications permanent immediately after the transaction has ended

**Domanda 4**

Completo

Punteggio max.: 4,00

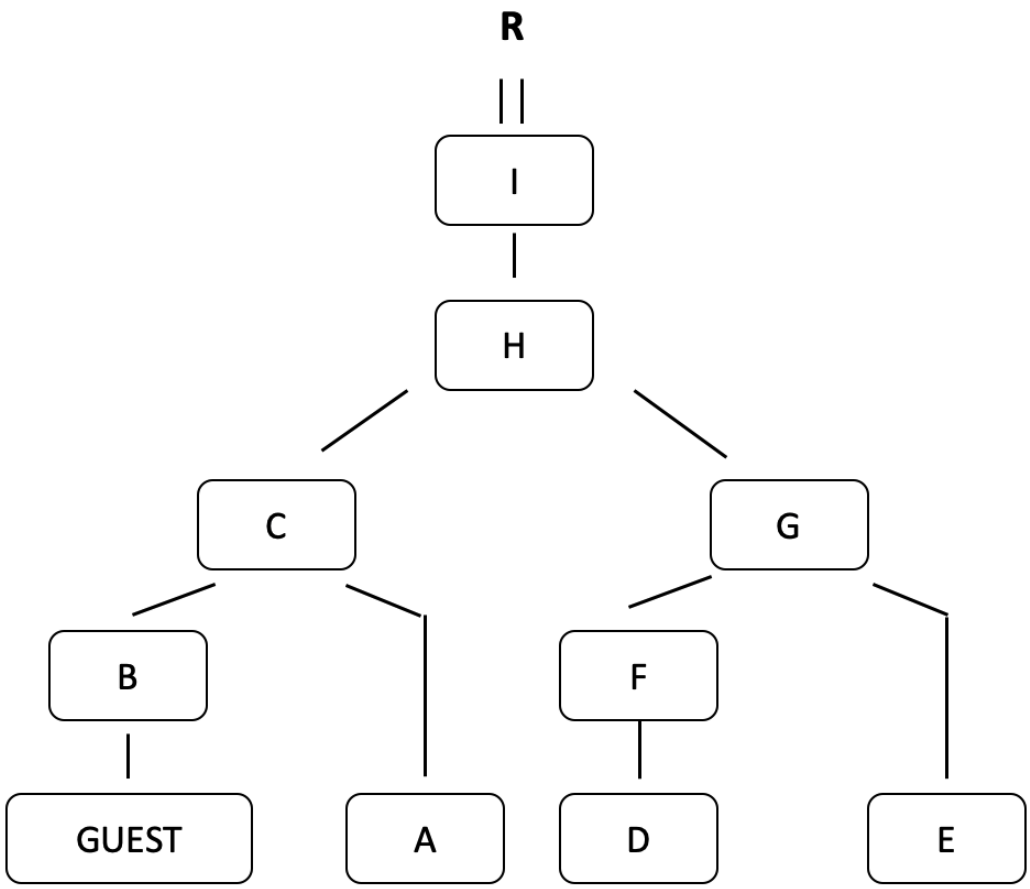
**Given the following relational tables:**

GUEST (GID, FirstName, LastName, BirthDate)
HOTEL (HID, Name, City, Region, #Stars)
STAY (GID, HID, StartDate, EndDate)

Select the name and the surname of guests born after 1990/01/01 who have stayed only in hotels located in the Piedmont region.

**Assignment for the exercise:**

*The following query tree graphically represents the requested algebraic query. You are requested to indicate, for each box in the query tree (i.e., A, B, C, D, E, F, G, H and I box), the relational table or the corresponding algebraic operator. Use the text box below to provide your solution. Note: each box in the query tree is associated with only one relational table or one algebraic operator.*




---

A. Stay S

B. select: BirthDATE> '1990/01/01'

C. Natural join

D. Hotel

E. Stay S1

F. select: Region <> 'Piedmont'

G. Natural join

H. Anti-semi-join: S.Gid = s1.Gid

I. projection: G.FirstName, G.LastName

**Draft Solution**

A. STAY1

B. Selection: BirthDate>1990/01/01

C. Theta-join: STAY1.GID=GUEST.GID or natural join

D. HOTEL

E. STAY2

F. Selection: Region <> ‘Piedmont’

G. Theta-join: STAY2.HID=HOTEL.HID or natural join

H. Anti semi-join / ANTI-join: GUEST.GID=STAY2.GID

I. Projection: FirstName, LastName

**Domanda 5**

Completo

Punteggio max.: 3,00

**Given the following relational tables**

```
Person (SSN, Name, Surname, DateOfBirth, Gender)
SerologicalTest (CodT, CommercialName, Brand, Reliability)
Building(CodB, City, Province Region, MaxCapacity)
UndergoTest (SSN, CodT, Date, CodB, Outcome)
```

**Write the following query in the SQL language:**

Find the name and surname of the male persons (Gender = “Male”) who have never undergone serological tests with a positive outcome (Outcome = “Positive”) in a building located in the city of Turin.

**Assignment for the exercise**

*Use the text box below to provide your solution.*

---

BLOCK A

SELECT SSN

FROM Building B, Undergotest U

WHERE B.CodB = U.CodB AND BCity = 'Turin' AND U.Outcome = 'Positive'

Main block

SELECT DISTINCT Name, Surname

FROM Person P

WHERE P.SSN NOT IN (BLOCK A) AND Gender = 'Male'

**Draft solution:**

```
SELECT DISTINCT Name, Surname
FROM Person
WHERE SSN NOT IN (
     SELECT SSN
     FROM UndergoTest UT, Building B
     WHERE UT.CodB = B.CodB AND
           Outcome = “Positive” AND
            City = “Turin”
) AND Gender = “Male”;
```

**Domanda 6**

Completo

Punteggio max.: 3,00

**Given the following relational tables**

```
Person (SSN, Name, Surname, DateOfBirth, Gender)
SerologicalTest (CodT, CommercialName, Brand, Reliability)
Building(CodB, City, Province Region, MaxCapacity)
UndergoTest (SSN, CodT, Date, CodB, Outcome)
```

**Write the following query in the SQL language:**

For each serological test brand, find the brand and the overall number of tests undergone from May 1st, 2020 to June 30th, 2020 to persons who have undergone at least two tests with the outcome “Positive” of that brand.

**Assignment for the exercise**

*Use the text box below to provide your solution.*

---

BLOCK A

(SELECT SSN, Brand

FROM UndergoTest U1, SeriologicalTest S1

WHERE U1.CodT = s1.CodT AND Outcome= Positive

GROUP BY SSN, Brand,

HAVING COUNT (*)>= 2) AS Pers

Main Block

SELECT Brand, COUNT(*)

FROM BLOCK A , SeriologicalTest S, UndergoTest U

WHERE Pers.SSN = U.SSN AND Pers.Brand =S.Brand AND S.CodT = U.CodT AND

Date >= 1/05/2020 AND Date <= 30/06/2020

GROUP BY Brand

DRAFT SOLUTION

**Sol 1: with correlation condition**

```
SELECT Brand, COUNT(*)
FROM SerologicalTest ST, UndergoTest UT
WHERE ST.CodT = UT.CodT AND Date >= 01/05/2020 AND Date <=
30/06/2020
AND SSN IN (
     Select SSN
     From UndergoTest UT2, SerologicalTest ST2
     where ST2.CodT = UT2.CodT and outcome='positive'
            and ST2.Brand = ST.Brand
     group by SSN
     having COUNT(*) > 1)
GROUP BY Brand
```

**Sol 2: with tuple constructor**

```
SELECT Brand, COUNT(*)
FROM SerologicalTest ST, UndergoTest UT
```

```
WHERE ST.CodT = UT.CodT AND Date >= 01/05/2020 AND Date <=
30/06/2020)
AND (SSN, Brand) IN (
Select SSN, Brand
     From UndergoTest UT2, SerologicalTest ST2
     where ST2.CodT = UT2.CodT and outcome='positive'
     group by SSN, Brand
     having COUNT(*) > 1)
GROUP BY Brand
```

**Sol 3: with table function**

```
SELECT Brand, COUNT(*)
FROM SerologicalTest ST, UndergoTest UT,
     (Select SSN, Brand
     From UndergoTest UT2, SerologicalTest ST2
     where ST2.CodT = UT2.CodT and outcome='positive'
     group by SSN, Brand
     having COUNT(*) > 1)AS TableTestBrand
WHERE ST.CodT = UT.CodT AND Date >= 01/05/2020 AND Date <=
30/06/2020)
AND TableTestBrand.Brand = ST.Brand AND TableTestBrand.SSN=UT.SSN
GROUP BY Brand
```

**Domanda 7**

Completo

Punteggio max.: 5,00

**Given the following relational tables**

```
Person (SSN, Name, Surname, DateOfBirth, Gender)
SerologicalTest (CodT, CommercialName, Brand, Reliability)
Building(CodB, City, Province Region, MaxCapacity)
UndergoTest (SSN, CodT, Date, CodB, Outcome)
```

**Write the following query in the SQL language:**

Considering only the buildings located in the Piedmont region (Region = “Piedmont”), find the dates of June 2020 at which the overall number of tests made in that building is maximal.

**Assignment for the exercise**

*Use the text box below to provide your solution.*

---

BLOCK A

( SELECT U.Date, B.CodB, COUNT()*AS N

FROM UndergoTest U, Building B

WHERE U.Date >= 01/05/21 AND U.Date<= 30/05/21 AND B.CodB = U.CodB AND B.Region= 'Piedomont'

GROUP BY B.CodB1) AS M

Main Block

SELECT B1.CodB, B1.Date

FROM BUILDING B, UNDERGOTEST U1

WHERE B1.CodB = U1.CodB AND B.Region= 'Piedmont' AND U1.Date>= 01/05/21 AND U1.Date <=31/05/21

GROUP BY B1.CodB, B1.Date

HAVING COUNT(*) = BLOCK B

**Draft solution:**

```
SELECT Date, B.CodB
FROM UndergoTest UT, Building B
WHERE Date >= 01/06/2020 AND Date <= 30/06/2020 AND
     UT.CodB = B.CodB AND
      Region = “Piedmont” AND
GROUP BY Date, B.CodB
HAVING COUNT (*) = (SELECT MAX(Test#)
                    FROM (SELECT Date, B1.CodB, COUNT(*) as
Test#
                          FROM UndergoTest UT1, Building B1
            WHERE Date >= 01/06/2020 AND
                          Date <= 30/06/2020 AND
                          UT1.CodB = B1.CodB AND
                           Region = “Piedmont” AND
            GROUP BY Date, B1.CodB) AS TF
                    WHERE TF.CodB=B.CodB
                  );
```

**Domanda 8**

Completo

Punteggio max.: 4,00

**Describe the Entity-Relationship diagram addressing the following specifications.**

You are requested to design the database for the management of company relocations.

The database must contain a list of vans suitable for relocations. The vans are identified by their plate, and they are characterized by their model, and by their volume in cubic meters if known. Of all the vans, some are authorized to transport special materials, and only for such vans, a list with the certifications of the special materials known to be allowed has to be stored.

The database must contain a list of warehouses, identified by a code and characterized by their address and the name of the company to which they belong.

You are requested to keep track of all the relocations made. The relocations are identified by the date and by the van with which they are made, and they are characterized by the name of the driver who carries it out. Each relocation is also characterized by the departure warehouse and departure time, and by the arrival warehouse and arrival time.

**Indications for solving the exercise**

Use the text box below to report the ER diagram in text form. Alternatively, you can use the drawing box to graphically represent the ER diagram.

---

ENTITY VAN

Primary Key: plate

Attribute: model, volume*

GENERALIZATION(p,e)

Parent entity: Van

Children entity: SpecialmaterialVan

Attribute: certification(1,N)

ENTITY WAREHOUSE

PrimaryKey: WCode

Attribute: Address, NomeCompany

ENTITY RELOCATION

PrimaryKey: Date

ForeignKey: plate (Van)

Attribute: DriverName

BINARY-RELATIONSHIP DEPARTURE

RELOCATION(1,1) WAREHOUSE (0,N)

Attribute: departureTime

BINARY-RELATIONSHIP ARRIVAL

RELOCATION (1,1) WAREHOUSE(0,N)

Attribute: arrivalTime

BINARY-RELATION BY

VAN(0,N) RELOCATION (1,1)

Entity VAN

ID: Plate

model, volume (0,1)

HIERARCHY (p,e)

children entity SPECIAL

certificationList (1,N)

Entity RELOCATION

internal ID interno: date

external ID: VAN's ID

driver, departureTime, arrivalTime (alternatively, the times can be added to the relationships RELOCATION-WAREHOUSE)

Entity WAREHOUSE

ID: WarehouseCode

address, companyName

Relazione CARRIED_OUT: VAN(0,N) – RELOCATION(1,1)

Relazione FROM: RELOCATION(1,1) – WAREHOUSE(0,N)

Relazione TO: RELOCATION(1,1) – WAREHOUSE(0,N)

**Domanda 9**

Completo

Punteggio max.: 3,00

**Describe the Entity-Relationship diagram addressing the following specifications.**

You are requested to design the database for the management of van repairs.

The database must contain a list of vans identified by their plate and characterized by the year of registration. Among the various types of vans, the capacity of the battery is known for those equipped with an electric motor.

The database must also include the list of repair shops, identified by a unique code and characterized by their address.

The repair shops can be in partnership with some companies. For each partner company, its VAT number, name, and possibly the list of telephone numbers are known. Note that each company can be in partnership with multiple repair shops and you are requested to keep track only of those companies having a partnership.

You are also requested to keep track of the repairs carried out over time on each van. The date, cost and duration in hours are known for each repair. Note that a van can undergo multiple repairs on the same day but in different repair shops. A repair shop can carry out at most one repair for the same van on the same day.

**Indications for solving the exercise**

Use the text box below to report the ER diagram in text form. Alternatively, you can use the drawing box to graphically represent the ER diagram.

---

ENTITY VAN

Primary Key: Plate

Attribute: yearofRegistration

GENERALIZATION(p,e)

ParentEntity: Van

Children entity; ElectricVan

Attribute: CapacityBattery

ENTITY Repair-shop

PrimaryKey: RCode

Attribute: Address

BINARY-RELATIONSHIP PARTNERSHIP

REPARI-SHOP(0,N) COMPANY(1,N)

ENTITY COMPANY

PrimaryKey: VatNumber

Attribute: name, telephone(0,N)

BINARY-RELATIONSHIP REPAIRED

REPAIR-SHOP(0,N) REPAY(1,1)

BINARY-RELATIONSHIP WHICH

VAN (0,N) REPAY(1,1)

ENTITY REPAY

PrimaryKey: Date

Foreign Key: RCode (REPAIR-SHOP), Plate (VAN)

Attribute: Cost, duration

Entity VAN

ID: Plate

- registrationYear

HIERARCHY (p,e)

children entity ELECTRIC

- batteryCapacity

Entity REPAIR_SHOP

ID: ShopCode

- address

Entity COMPANY

ID: VATnumber

- name, phoneNumbers (0,N)

Relationship PARTNERSHIP: REPAIR_SHOP(0,N) – COMPANY(1,N)

Relationship REPAIR: REPAIR_SHOP(0,N) – VAN(0,N) – TIME(1,N), attributes: price, duration

(alternatively, REPAIR entity with external IDs from repair_shop and van, and internal id: date)

Entity TIME

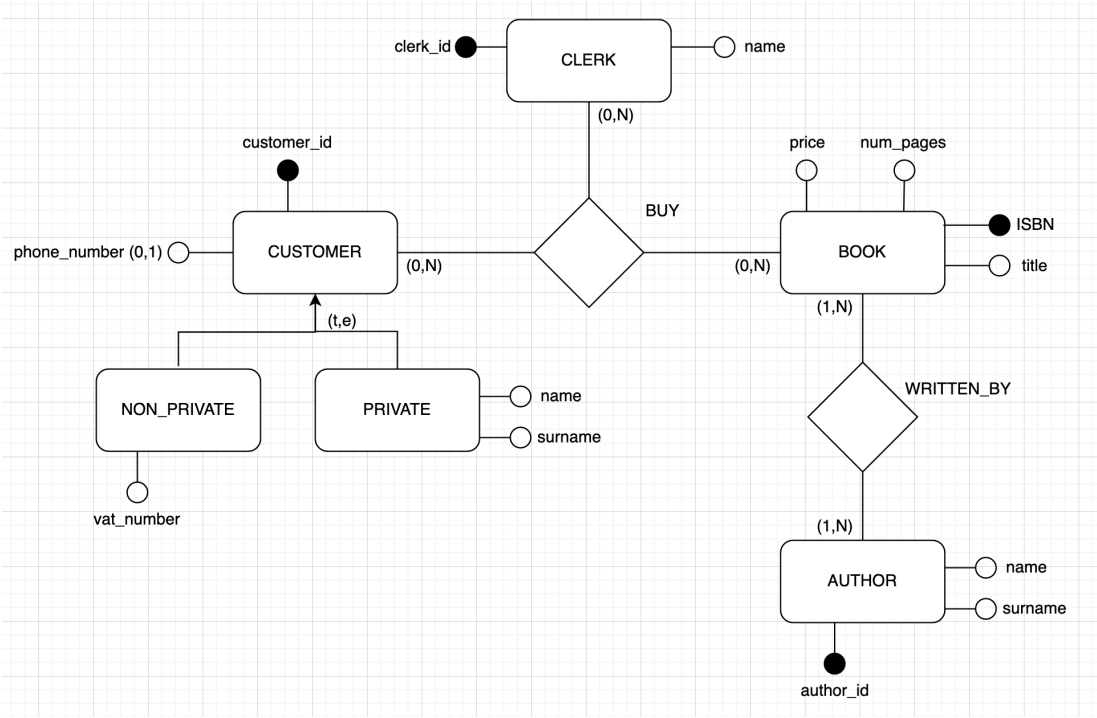
ID: date

**Domanda 10**

Completo

Punteggio max.: 3,00

**Given the following Entity-Relationship diagram**




You are required to:

- Provide a normalized relational logical schema for the same database (N.B. It is not mandatory to report the restructured E-R diagram)
- Define referential integrity constraints for 2 relationships of your choice among those defined in the conceptual schema

**Assignment**

Use the text box below to provide your solution.

---

AUTHOR(Name, Surname, AuthorID_)

BOOK (ISBN_, title, price, num-pages)

WRITTEN_BY(AuthorId_, ISBN_)

CLERK (ClerkId_, name)

CUSTOMER(CustomerId_, phone-number*)

BUY(ClerkId_, CustomerId_, ISBN_)

Non-private(CustomerId_, vat_number)

Private(ClerkId_, name, surname)

REFERENTIAL INTEGRITY CONSTRAINT

Written-by(AuthorID) REFERENCES Author(AuthorID)

Written-by(ISBN) REFERENCES Book(ISBN)

BUY(ClerkId) REFERENCES Clerk(Clerk-id)

BUY(CustomerID) REFERENCES Customer(Customer-id)

BUY(ISBN) REFERENCES BOOK(ISBN)

AUTHOR(author_id, name, surname)

BOOK(ISBN, title, num_pages, price)

CUSTOMER(customer_id, phone_number*, type, name*, surname*, vat_number*)

CLERK(clerk_id, name)

BUY(customer_id, clerk_id, ISBN)

WRITTEN_BY(author_id, ISBN)

1) WRITTEN_BY (author_id) REFERENCES AUTHOR(author_id)
   WRITTEN_BY (ISBN) REFERENCES BOOK(ISBN)

2) BUY(customer_id) REFERENCES CUSTOMER(customer_id)
   BUY (clerk_id) REFERENCES CLERK(clerk_id)
   BUY (ISBN) REFERENCES BOOK(ISBN)

**Domanda 11**

Completo

Punteggio max.: 3,00

**The following relational schema is given (primary keys are underlined):**

MARKET_BASKET (BasketCode, ItemCode, NumberofPieces)
ITEM_PRICE (ItemCode, CostPerPiece)
BASKET_TOTALPRICE_NOTIFICATION (RequestCode, BasketCode, TotalBasketCost)
LOYALTY_CARDS (LoyaltyCardCode, TotalAmount)
REQUEST_FOR_CALCULATION_OF_TOTALBASKETPRICE (RequestCode, BasketCode, LoyaltyCardCode)

Write the trigger to manage the following activities on an online shopping site. The calculation of the total cost of a market basket is requested (insertion of a record in the REQUEST_FOR_CALCULATION_OF_TOTALBASKETPRICE table).

The calculation of the total price of the basket must consider the items in the basket, the number of pieces per item, and the one-piece price of each item. The MARKET_BASKET table contains, for each basket, the items contained in the basket and the number of pieces per item. The ITEM_PRICE table contains the one-piece price of each item.

Once the total price of the market basket has been computed, a new record must be inserted in the BASKET_TOTALPRICE_NOTIFICATION table with the calculated information. Then, the total amount for the customer who requested the calculation of the market basket price must be updated in the LOYALTY_CARDS table (attribute LoyaltyCardCode in the REQUEST_FOR_CALCULATION_OF_TOTALBASKETPRICE table).

**Indications for carrying out the exercise**

*Given the following incomplete solution of the trigger, you are asked to complete **Part A** in bold by specifying the body of the trigger. Use the text box below to provide your solution.*

create or replace trigger CalculationOfTotalBasketPrice
after insert on REQUEST_FOR_CALCULATION_OF_TOTALBASKETPRICE
for each row
**Part A**

---

```
DECLARE
Summed Number;

BEGIN
    SELECT SUM(NumberOfPieces*CostPerPiece) INTO Summed
    FROM Market_Basket M, Item_Price I
    WHERE M.ItemCode = I.ItemCode AND M.BasketCode=        :New.BasketCode;

    INSERT INTO  BASKET_TOTALE_PRICE_NOTIFICATION(RequestCode, BasketCode, TotalBasketCost) VALUES (:NEW.RequestCode, :New.BasketCode, summed)
UPDATE LOYALTY_CARDS(LoyaltyCode, TotalAmount)
  SET TotalAmount = TotalAmount+x
  WHERE LoyaltyCardCode = :NEW.LoyaltyCardCode;
```

END;

**Draft Solution -** Part A:

```
declare
X number;
begin

select SUM(NumberofPieces*CostPerItem) INTO X
from MARKET_BASKET MB,ITEM_PRICE IP
where BasketCode = :NEW.BasketCode AND MB.ItemCode = IP.ItemCode;

INSERT INTO BASKET_TOTALPRICE_NOTIFICATION (RequestCode, BasketCode,
TotalBasketPrice) values (:New.RequestCode, :NEW.ItemCode, X);

UPDATE LOYALTY_CARDS
SET TotalAmount = TotalAmount + X
WHERE LoyaltyCardCode = :NEW.LoyaltyCardCode;

END;
```