POLITECNICO
DI TORINO

Introduction to databases

Exam 2021-06-28

[redacted]

Iniziato	lunedì, 28 giugno 2021, 08:21
Terminato	lunedì, 28 giugno 2021, 09:51
Tempo impiegato	1 ora 30 min.
Valutazione	**24,00** su un massimo di 31,00 (**77**%)

Domanda 1

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Given the following PHP statement:

(where the variable $results stores the output of a SQL statement)

```
$var = mysql_fetch_assoc($result);
```

- (a)
  none of the other answers are correct
- (b)
  $var is an integer
- (c)
  the elements of the array $var can be accessed both with a numeric index and with the key corresponding to the attribute name of the returned table
- (d)
  $var is an associative array ✓

Risposta corretta.

La risposta corretta è:
$var is an associative array

Domanda 2

Risposta corretta

Punteggio ottenuto 1,00 su 1,00

Let V1(A,B) be a view that has been previously defined.
The command

```
CREATE VIEW V2
AS SELECT DISTINCT B FROM V1;
```

- (a)

  is correct if B is the primary key of V1

- (b)
  is incorrect because SQL does not allow defining views based on other views

- (c)
  is always correct
  ✓

- (d)
  none of the other answers are correct

Risposta corretta.

La risposta corretta è:
is always correct

Domanda 3

Risposta errata

Punteggio ottenuto 0,00 su 1,00

The SQL command

```
REVOKE SELECT, UPDATE(Q)
ON TABLE SIP
FROM Jim
WITH GRANT OPTION;
```

- (a)
  revokes authorization to read the data in table SIP, and to update column Q in table SIP from user Jim, also revoking this privilege from all other users that have possibly received it from Jim
  ✗
- (b)
  is incorrect because it is not possible to specify WITH GRANT OPTION within a REVOKE command
- (c)
  none of the other answers are correct
- (d)
  revokes authorization to update column Q in table SIP from user Jim

Risposta errata.

La risposta corretta è:
is incorrect because it is not possible to specify WITH GRANT OPTION within a REVOKE command

Domanda 4

Completo

Punteggio ottenuto 4,00 su 4,00

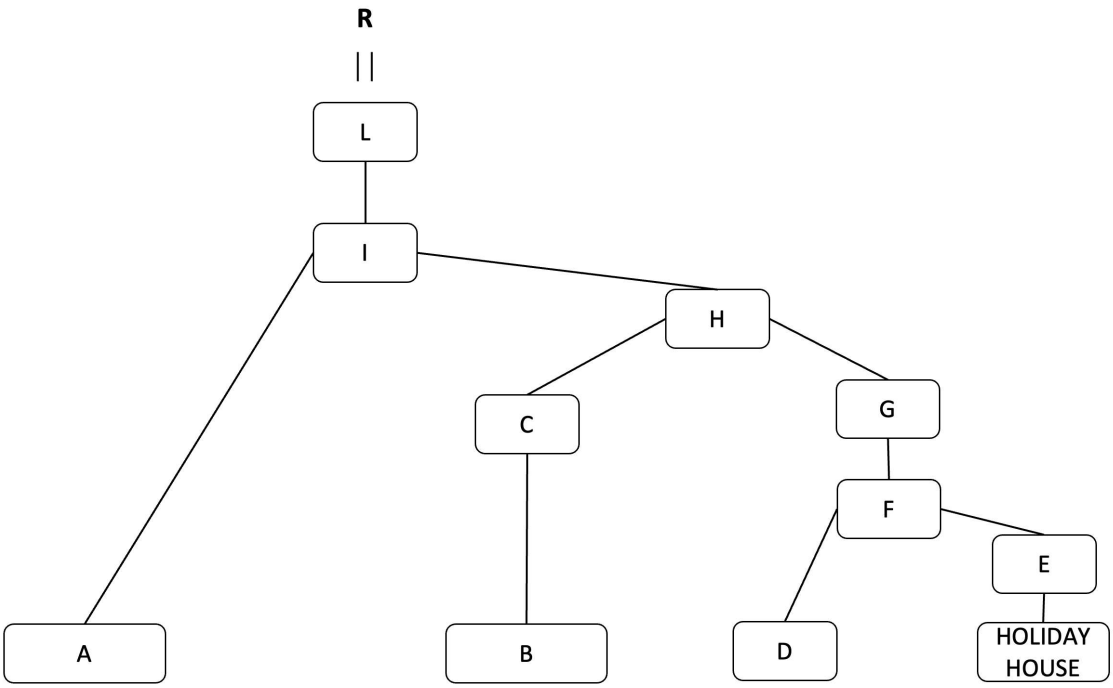
Given the following schema:

CUSTOMER (SSN, Name, Surname, BithDate, City)
HOLIDAY-HOUSE (HID, Name, Type, Address, City, WeeklyRentPrice)
REVIEW (SSN, HID, Date, Text, Score)

Display name and surname of customers who reviewed only holiday houses of type “studio”.

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, 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, CUSTOMER

B, REVIEW

C, PROJECTION(SSN)

D, REVIEW

E. SELECTION(Type<>"studio")

F, NATURAL JOIN

G, PROJECTION(SSN)

H, DIFFERENCE

I, NATURAL JOIN

L, PROJECTION (Name,Surname)

Draft solution

A= CUSTOMER C1
B= REVIEW R1
C= Projection: SSN
D= REVIEW R2
E= Selection: Type<>Studio'
F= Natural join or Theta-join(semi-join): R2.HID= HOLIDAY-HOUSE.HID
G= Projection: SSN
H= Difference
I= Natural join or Theta-join on C1.SSN = R1.SSN
L = Projection: Name, Surname

Commento:

Domanda 5

Completo

Punteggio ottenuto 2,50 su 3,00

Given the following relations (primary keys are underlined, whereas the fields that can take the NULL value are denoted as "*"):

JUDGE (SSN, Name, Surname, BirthDate)

COURTHOUSE (CourthouseId, CourthouseName, City)

COURTROOM (CourthouseId, CourtroomId, CourtroomName)

CASE (CaseId, CaseType, StartingDate, EndingDate*, SSN)

HEARING (CourthouseId, CourtroomId, Date, StartingTime, EndingTime, CaseId)

For each judge who has never chaired any defamation case (CaseType='Defamation') in 2020, find SSN, surname, and number of different case types chaired by the judge.

SELECT SSN, Surname, COUNT(DISTINCT CaseType) As DifferentTypeCase#

FROM JUDGE J, CASE C

WHERE J.SSN=C.SSN

```
    AND J.SSN NOT IN
    (
        SELECT SSN
        FROM CASE
        WHERE CaseType="Defamation" AND StartingDate>='2020/01/01' AND
StartingDate<='2020/12/31"
    )
GROUP BY J.SSN, J.Surname
```

Draft Solution

```
SELECT SSN, Surname, COUNT(DISTINCT CaseType)
FROM JUDGE J, CASE C
WHERE J.SSN = C.SSN and J.SSN NOT IN
   (SELECT C2.SSN
    FROM CASE C2, HEARING H
    WHERE Date>=1/1/2020 and Date<=31/12/2020 and CaseType = 'Defamation' And
H.CaseId = C2.CaseId)
)
GROUP BY J.SSN, Surname;
```

Commento:
Table HEARING is required in the NOT IN block

Domanda 6

Completo

Punteggio ottenuto 2,25 su 3,00

Given the following relations (primary keys are underlined, whereas the fields that can take the NULL value are denoted as "*"):

JUDGE (SSN, Name, Surname, BirthDate)

COURTHOUSE (CourthouseId, CourthouseName, City)

COURTROOM (CourthouseId, CourtroomId, CourtroomName)

CASE (CaseId, CaseType, StartingDate, EndingDate*, SSN)

HEARING (CourthouseId, CourtroomId, Date, StartingTime, EndingTime, CaseId)

Considering the already ended cases chaired by a judge who (throughout his career) has chaired hearings in no more than 5 different courthouses, find the case type and the date of its first hearing.

/* Uso IN e non NOT IN perché sembra chiedere come se ha tenuto almeno un udienza in una courthouses*/

```
SELECT CA.SSN, HE.Date, CA.CaseType
FROM CASE CA, HEARING HE
WHERE CA.CaseId=HE.CaseId
   CA.SSN IN
   (
   SELECT SSN
   FROM HEARING H, CASE C
   WHERE H.CaseId=C.CaseId
   GROUP BY C.SSN
   HAVING COUNT(DISTINCT CourthouseID)<=5
   )
   AND CA.EndingDate IS NOT NULL
   AND HE.Date =
      ( SELECT MIN(Date)
        FROM HEARING HE1, CASE CE1
        WHERE HE1.CaseID=CE1.CaseID
              AND CE1.SSN=CA.SSN
        AND CE1.EndingDate IS NOT NULL
       )
```

```
SELECT CaseType, MIN(Date)
FROM CASE C, HEARING H
WHERE C.EndingDate IS NOT NULL AND C.CaseId = H.CaseId
AND C.SSN IN
   (SELECT C2.SSN
    FROM CASE C2, HEARING H2
    WHERE C2.CaseId = H2.CaseId
    GROUP BY C2.SSN
    HAVING COUNT(DISTINCT CourthouseId)<=5)
)
GROUP BY C.CaseId, CaseType;
```

Commento:
External Select is wrong partially

The correlation condition is wrong

Domanda 7

Completo

Punteggio ottenuto 2,75 su 5,00

Given the following relations (primary keys are underlined, whereas the fields that can take the NULL value are denoted as “*”):

JUDGE (SSN, Name, Surname, BirthDate)

COURTHOUSE (CourthouseId, CourthouseName, City)

COURTROOM (CourthouseId, CourtroomId, CourtroomName)

CASE (CaseId, CaseType, StartingDate, EndingDate*, SSN)

HEARING (CourthouseId, CourtroomId, Date, StartingTime, EndingTime, CaseId)

For each judge who has chaired hearings in all the courthouses, where in each of them at least 10 divorce cases (CaseType=’Divorce’) were held, find the surname, surname, and birthdate of the judge.

/* Errore nel testo, considero name e surname, per come è scritto il testo ho interpretato "giudice che ha partecipato a tutte le corthouse e inoltre
in ognuna di esse deve aver fatto almeno 10 casi di divorzio (non per forza diversi) */

```
SELECT Name, Surname, BirthDate
FROM JUDGE
WHERE SSN IN
   (
   SELECT SSN
   FROM (SELECT SSN, CourthouseID
    FROM HEARING H, CASE C
    WHERE H.CaseID=C.CaseID
            AND C.CaseType="Divorce"
```

```
GROUP BY SSN, CourthouseID
HAVING COUNT(*)>=10
) AS res
 GROUP BY SSN
 HAVING COUNT(*) =
  (  SELECT COUNT(*)
     FROM COURTHOUSE
   )
)
```

Draft Solution

```
SELECT DISTINCT J.Name, J.Surname, J.BirthDate
FROM JUDGE J, HEARING H, CASE C
WHERE J.SSN = C.SSN AND H.CaseId = C.CaseId
AND H.CourtHouseId IN
   (SELECT H2.CourtHouseId
    FROM HEARING H2, CASE C2
    WHERE H2.CaseId = C2.CaseId
         AND C2.CaseType = 'Divorce'
    GROUP BY H2.CourtHouseId
    HAVING COUNT(DISTINCT H2.CaseId) >= 10
    )
GROUP BY J.SSN, J.Surname, J.BirthDate
HAVING COUNT(DISTINCT H.CourtHouseId) =
   (SELECT COUNT(DISTINCT CourtHouseId)
    FROM COURTROOM CR
    WHERE CH.CourtHouseId IN
       (SELECT H3.CourtHouseId
       FROM CASE C3, HEARING H3
       WHERE H3.CaseId = C3.CaseId
       AND C3.CaseType = 'Divorce'
       GROUP BY H3.CourtHouseId
          HAVING COUNT(DISTINCT H3.CaseId) >= 10)
    );
```

Commento:
The second part of the query is wrong, while the first contains some errors

Domanda 8

Completo

Punteggio ottenuto 1,50 su 3,00

Describe the Entity-Relationship diagram addressing the following specifications.

The National Institute of Astrophysics is required to design a database collecting information relating to the experimental sessions executed at the Italian astronomical observatories.

- Astronomical observatories are identified by their location, they are characterized by a list of telephone numbers, and they include some telescopes.
- The telescopes are identified by a number, unique within the observatory to which they belong, and they are characterized by the year of construction.
- The researchers, whose name, surname, and social security number are known, are affiliated with one or more observatories, and they can use the telescopes for experimental sessions. For each observatory and affiliated researcher, the latest period of affiliation is known (start date, end date). Please note that the affiliation can still be ongoing, hence the end date could be unknown. An observatory can have many affiliated researchers.
- The database must record the different experimental sessions executed at the telescopes by researchers. For each experimental session, the date, the telescope used, the researcher who conducts it, and the name of the research project are known. A researcher can carry out an experimental session with the same telescope on different days, a researcher can carry out several experimental sessions with different telescopes on the same day. A telescope cannot be used on the same day for more than one experimental session.

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: Observatory:

Int ID: Location

Attr: Phone#(0,n)

ENTITY: Telescope

Int ID: Number

Ext ID: Location REFERENCES Observatory(Location)

Attr: YearCostr

RELATIONSHIP: HAS_TELESCOPE

Observatory(1,N)-Telescope(1,1)

ENTITY: RESEARCHES

Int ID: SSN

Attr: Name, Surname

ENTITY: TIME

Int ID: StartDate

RELATIONSHIP: HAS_RESEARCHER

Observatory(1,N)-RESEARCHES(1,N)- TIME(1,N)

Attr: EndDate*

ENTITY: Exmperimental_session

IntID: StartDate

ExtID: (Number,Location) REFERENCES Telescope(Number,Location)

Attr: NameResearch, SSN REFERENCES RESEARCHES(SSN)

RELATIONSHIP:

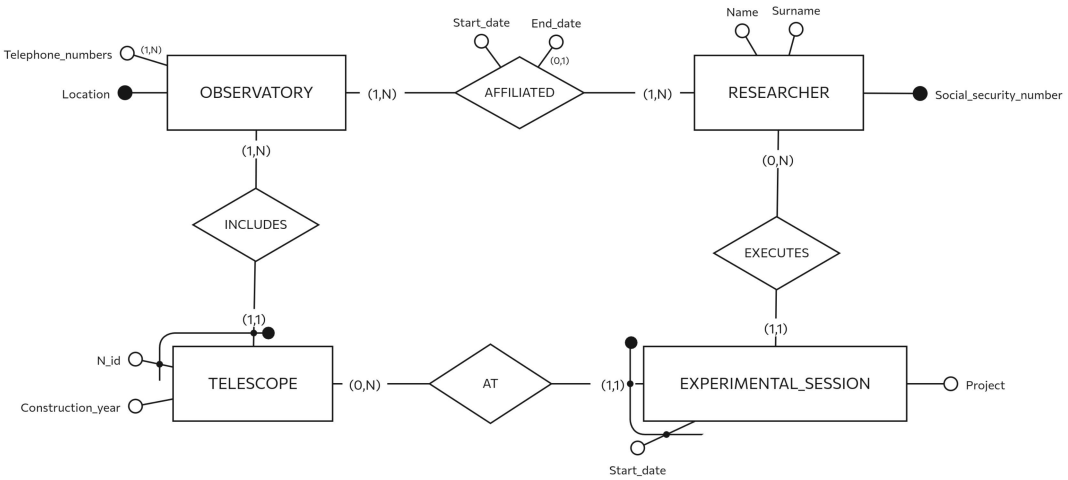
Experimental_session(1,1)-RESEARCHES(0,N)

Draft solution

```
/* Entities */
entity OBSERVATORY {
   Location (id),
   Telephone_number (1..N)
}
entity TELESCOPE {
   N_id (id),
   Location (external_id from OBSERVATORY),
   Construction_year
}
entity RESEARCHER {
   Social_security_number (id),
   Name,
   Surname
```

}
entity **SESSIONE_SPERIMENTALE** {
Start_date (id),
N_id (external_id from **TELESCOPE**),
Project
}

Relationships:
relationship **INCLUDES** (
OSSERVATORY: 1..N,
TELESCOPE: 1..1 external_id
)
relationship **AT** (
TELESCOPE: 0..N,
EXPERIMENTAL_SESSION: 1..1 external_id
)
relationship **EXECUTES** (
EXPERIMENTAL_SESSION: 1..1,
RESEARCHER: 0..N
)
relationship **AFFILIATED**(
OBSERVATORY: 1..N,
RESEARCHER: 1..N
- Start_date
- End_date (0,1)
)



Commento:
entity time should not be used in this solution

Domanda 9

Completo

Punteggio ottenuto 3,00 su 4,00

Describe the Entity-Relationship diagram addressing the following specifications.

You are required to create a database for the management of the booking system of a multiplex cinema.

- The cinema has some thematic halls, identified by a number. For each hall, its name, its capacity and the list of movie genres associated with the hall are known. A movie genre can be associated with different halls.
- The movies at the cinema are identified by a unique code and are characterized by their title, genre, and year of release.
- Each hall hosts screenings of different movies. The screening of a movie can take place in the same room at different times (date and start time), and in different rooms, even at the same time. For each screening, the database must record the list of customers watching it.
- For each customer, their name, social security number, telephone number and date of birth are known.
- Some customers have a loyalty card, which grants them some discounts. For such customers, the card number and the expiration date of the card are known.

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: HALL
IntID: Number

Attr: Name, Capacity

ENTITY: Genre

IntID: NameGenre

RELATIONSHIP: HasGenre

Genre(1,N) - HALL(1,N)

ENTITY: MOVIE

IntID: CodeM

Attr: Title,YearRelease, NameGenre REFERENCES GENRE(NameGenre)

RELATIONSHIP:

MOVIE(1,1)-GENRE(1,N)

ENTITY: SCREENING

Int ID: Date, StartTime

Ext ID: NumberHall REFERENCES HALL(Number), CodeM REFERENCES MOVIE(CodeM)

RELATIONSHIP: SCREEN-WATCHED-BY

SCREENING(0,N)-SSN(0,N)

ENTITY: CUSTOMER

IntID:SSNCust

attr: Name, Phone#, DateOfBirth

CUSTOMER is Father, HIERARCLY(p,s)

ENTITY child: LOYALTY-MEMBER

Attr:Card#, ExpirationDateCard

Draft Solution

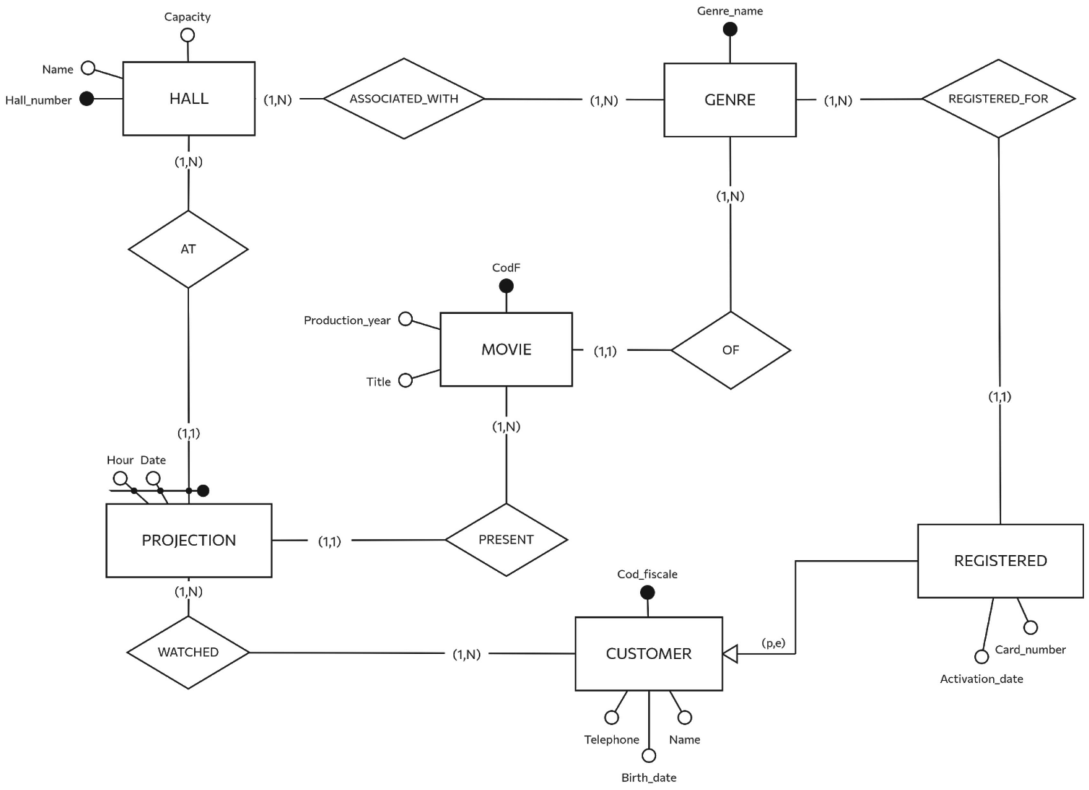
```
/* Entities */
entity HALL {
   Hall_number (id),
   Name,
   Capacity
}
entity GENRE {
   Genre_name (id)
}
entity PROJECTION{
   Date (id),
   Hour (id),
   Hall_number (external_id from HALL)
}
```

```
entity CUSTOMER {
    Social_security_number (id),
    Name,
    Telephone,
    Birth_date
}
entity REGISTERED {
    Card_number,
    Activation_date
}
entity MOVIE {
    CodF (id),
    Production_year,
    Title
}


/* Relationships */
relationship ASSOCIATED_WITH (
    HALL: 1..N,
    GENRE: 1..N
)
relationship AT (
    HALL: 1..N,
    PROJECTION: 1..1
)
relationship WATCHED (
    PROJECTION: 1..1 external_id,
    CUSTOMER: 1..N
)

relationship OF (
    FILM:1..1,
    GENERI: 1..N
)
relationship PRESENTS (
   MOVIE: 1..N,
    PROJECTION: 1..1
)

/* Generalizations */
CUSTOMER <= {
    REGISTERED
} (partial, exclusive)
```



Commento:

in screening the identifier of movie should not be part of the id

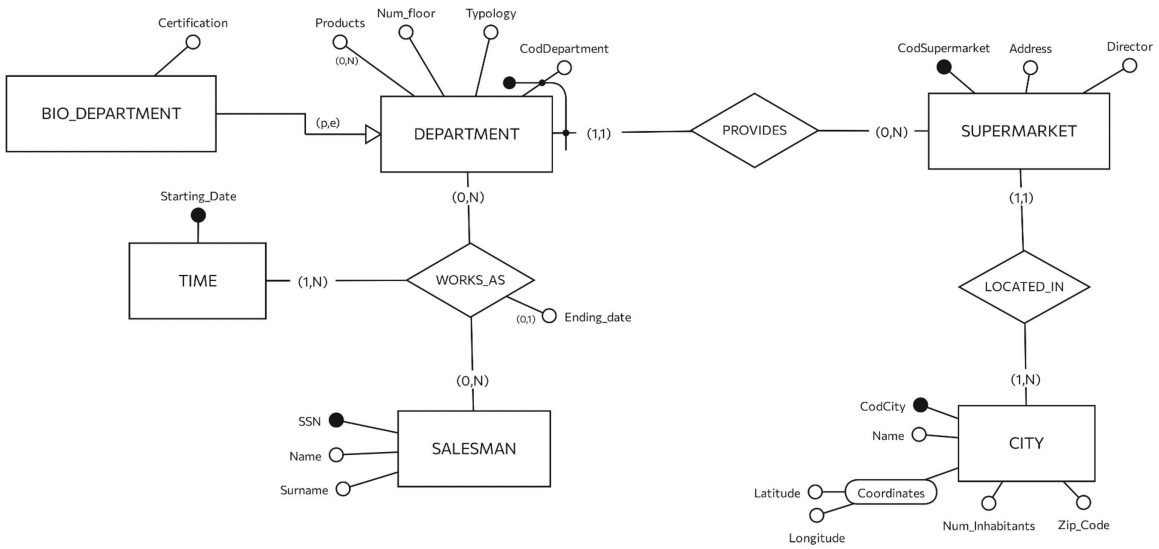
some cardinality errors

Domanda 10

Completo

Punteggio ottenuto 3,00 su 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.

SALESMAN(SSN,Surname,Name)

CITY(CodCity, Name, Latitude, Longitude, Numb_InHabitants, Zip_Code)

SUPERMARKET(CodSuperMarket, Address, Director, CodCity)

DEPARTMENT(CodDepartment,CodSuperMarket, Typology,, Num_floor,CertificationIfBioDep*)

PRODUCTS(CodDepartment, CodSuperMArket, NameProduct)

WORK_AS(SSN,StartingDate,CodDepartment,CodSuperMarket, Ending_date*)

1)

SUPERMARKET(CodCity) REFERENCES CITY(CodCity)

2)

DEPARTMENT(CodSuperMarket) REFERENCES SUPERMARKET(CodSuperMarket)

Draft Solution

SUPERMARKET(CodSupermarket, Address, Director, CodCity)
CITY(CodCity, Name, Num_Inhabitants, ZIP_Code, Latitude, Longitude)
DEPARTMENT(CodDepartment, CodSupermarket, Num_floor, Lanes, DepartmentType, Certification*)
PRODUCTS(CodProduct, CodSupermarket, CodDepartment)
SALESMAN(SSN, Name, Surname)
WORK_AS(SSN, CodSupermarket, CodDepartment, Starting_Date, Ending_Date*)

Constraints:

SUPERMARKET(CodCity) REFERENCES CITY(CodCity)

DEPARTMENT(CodSupermarket) REFERENCES SUPERMARKET(CodSupermarket)WORK_AS(SSN) REFERENCES SALESMAN(SSN)

WORK_AS(CodSupermarket, CodDepartment) REFERENCES DEPARTMENT(CodCodSupermarket, CodDepartment)

Commento:

Domanda 11

Completo

Punteggio ottenuto 3,00 su 3,00

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

SPORT_ACTIVITY (ActivityCode, Description, NumberOfAvailablePlaces)
EQUIPMENT (ActivityCode, EquipmentCode, Cost)
ENROLLMENT_NOTIFICATION (SSN, ActivityCode, TotalCost)
SPORT_ACTIVITY_ENROLLMENT_REQUEST (RequestCode, SSN, ActivityCode)

Write the trigger to manage the following activities related to the enrollment in sport activities. The SPORT_ACTIVITY table contains the list of sport activities and the number of places still available for each sport activity. The EQUIPMENT table contains, for each sport activity, the list of all the equipment required for the sport activity and the cost for each piece of equipment.

When there is a new enrollment request for sport activity, a new tuple is inserted into the SPORT_ACTIVITY_ENROLLMENT_REQUEST table.

If there are no places available for the requested sport activity, the trigger ends with an error. If there are available places, the cost of the sport activity must be calculated. The calculation must

be made by considering all the equipment items required for the sport activity, and the cost of each piece of equipment.

A new record must be inserted into the ENROLLMENT_NOTIFICATION table with the calculated information. The SPORT_ACTIVITY table must then be updated by decreasing by one unit the number of places available for the sport activity.

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 trigger execution mode (before or after) and Part B in bold by specifying the body of the trigger. Use the text box below to provide your solution.

Use the raise_application_error (....) function to raise an error. It is not required to specify the parameters passed to the function.

```
create or replace trigger ActivityTotalCost
Part A insert on SPORT_ACTIVITY_ENROLLMENT_REQUEST
for each row
Part B
```

A| AFTER

B|

```
DECLARE
Available Number;
TotalCost Number;

BEGIN

SELECT NumberOfAvailablePlaces INTO Available
FROM SPORT_ACTIVITY
WHERE ActivityCode= :NEW.ActivityCode;

if(Available=0) then
   raise_application_error();
end if;
```

```
SELECT SUM(Cost) INTO TotalCost
FROM EQUIPMENT
WHERE ActivityCode= :NEW.ActivityCode;

INSERT INTO ENROLLMENT_NOTIFICATION
VALUES( :NEW.SSN, :NEW.ActivityCode, TotalCost);

UPDATE TABLE SPORT_ACTIVITY
SET NumberOfAvailablePlaces= NumberOfAvailablePlace -1
WHERE ActivityCode= :NEW.ActivityCode;

END;
```

Draft Solution

Blocco A: after
Blocco B:

```
Declare
X number
Begin

select NumberOfAvailablePlaces INTO X
from SPORT_ACTIVITY
where ActivityCode = :NEW.ActivityCode;

IF NumberOfAvailablePlaces =0 then
Raise_application_error (….)
End if;

SELECT sum(Cost) INTO X
FROM EQUIPMENT
WHERE ActivityCode = :NEW.ActivityCode;

Update SPORT_ACTIVITY
Set NumberOfAvailablePlaces = NumberOfAvailablePlaces -1
Where ActivityCode = :new.ActivityCode;

INSERT INTO NOTIFICA_ISCRIZIONE (SSN, ActivityCode, TotalCost)
VALUES (:NEW.SSN, :NEW.ActivityCode, X);

end;
```

Commento: