Bases de Datos relacionales

2023-01-17

Introducción

Iniciemos creando una conexión al archivo SQL

```
mydb <- dbConnect(RSQLite::SQLite(), "my-db.sqlite")
dbListTables(mydb)
```

```
## [1] "company_mast"   "customer"        "emp_department" "
## [5] "item_mast"      "orders"          "salesman"
```

```
dbListFields(mydb, "salesman")
```

```
## [1] "salesman_id" "name"          "city"          "commissio
```

Ejecutamos la primer consulta

```
dbGetQuery(mydb, 'SELECT * FROM salesman')
```

```
##   salesman_id        name     city commission
## 1        5001 James Hoog New York       0.15
## 2        5002 Nail Knite    Paris       0.13
## 3        5005   Pit Alex   London       0.11
## 4        5006    Mc Lyon    Paris       0.14
## 5        5007  Paul Adam     Rome       0.13
## 6        5003 Lauson Hen San Jose       0.12
```

Ejercicios

1. From the following tables write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust_name and city. (salesman, customer)
2. From the following tables write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city. (orders, customer)
3. From the following tables write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission. (customer, salesman)
4. From the following tables write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman, commission. (customer, salesman)

5. From the following tables write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission. (customer, salesman)

6. From the following tables write a SQL query to find the details of an order. Return ord_no, ord_date, purch_amt, Customer Name, grade, Salesman, commission. (orders, customer, salesman)

7. Write a SQL statement to join the tables salesman, customer and orders so that the same column of each table appears once and only the relational rows are returned. (orders, customer, salesman)

8. From the following tables write a SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer_id. (customer, salesman)

9. From the following tables write a SQL query to find those customers with a grade less than 300. Return cust_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer_id. (customer, salesman)

10. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to determine whether any of the existing customers have placed an order or not. (orders, customer)

11. SQL statement to generate a report with customer name, city, order number, order date, order amount, salesperson name, and commission to determine if any of the existing customers have not placed orders or if they have placed orders through their salesman or by themselves. (customer, orders, salesman)

12. Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers. (customer, salesman)

13. From the following tables write a SQL query to list all salespersons along with customer name, city, grade, order number, date, and amount. (customer, salesman, orders)

14. Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customer. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier. (customer, salesman, orders)

15. For those customers from the existing list who put one or more orders, or which orders have been placed by the customer who is not on the list, create a report containing the customer name, city, order number, order date, and purchase amount. (customer, orders)

16. Write a SQL statement to generate a report with the customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who neither is on the list nor has a grade. (customer, orders)

17. Write a SQL query to combine each row of the salesman table with each row of the customer table. (salesman, customer)

18. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for all customers and vice versa for that salesperson who belongs to that city. (salesman, customer)

```
dbDisconnect(mydb)
```