DATA ANALYST

ASSESSMENT

SQL Queries

Important Note:

All the dates in the assignment follow mm/dd/yyyy nomenclature. All the answeres are to be given in the query form. Do not list the result of the query. There can be more than one way of writing a query. As long as the query is logically correct it will be accepted.

|  |  |
| --- | --- |
| Employee | Location |
| |  |  | | --- | --- | | EmpID | EmpName | | 13 | Jason | | 8 | Alex | | 3 | Ram | | 17 | Babu | | 25 | Johnson | | |  |  | | --- | --- | | EmpID | EmpLoc | | 13 | San Jose | | 8 | Los Angeles | | 3 | Pune, India | | 17 | Chennai, India | | 39 | Bangalore, India | |

1. Write a query to get a location of every Employee

SELECT Employee.EmpName, Location.EmpLoc

FROM Employee

JOIN Location

ON Employee.EmpID = Location.EmpID;

1. Write the above query using table alias

SELECT t1.EmpName, t2.EmpLoc

FROM Employee t1

JOIN Location t2

ON t1.EmpID = t2.EmpID;

1. Name the employees that are not living in India.

SELECT t1.EmpName

FROM Employee t1

JOIN Location t2

ON t1.EmpID = t2.EmpID

WHERE EmpLoc NOT LIKE “%india%” ;

|  |
| --- |
| employee |
| |  |  | | --- | --- | | **employee\_name** | **employee\_location** | | Joe | New York | | Sunil | India | | Alex | Russia | | Albert | Canada | | Jack | New York | |

1. Find out which employees are from the same location as the employee named Joe.

SELECT employee\_name

FROM employee

WHERE employee\_location = (SELECT employee\_location

FROM employee WHERE employee\_name = ‘Joe’);

1. Get the names of employee in lower case.

SELECT LOWER(employee\_name) AS employee\_name\_lower

FROM employee;

|  |  |
| --- | --- |
| Salesperson | Customer |
| |  |  |  |  | | --- | --- | --- | --- | | ID | Name | Age | Annual Salary | | 1 | Abe | 61 | 140000 | | 2 | Bob | 34 | 44000 | | 5 | Chris | 34 | 40000 | | 7 | Dan | 41 | 52000 | | 8 | Ken | 57 | 115000 | | 11 | Joe | 38 | 38000 | | |  |  |  |  | | --- | --- | --- | --- | | ID | Name | City | Industry Type | | 4 | Samsonic | pleasant | J | | 6 | Panasung | oaktown | J | | 7 | Samony | jackson | B | | 9 | Orange | Jackson | B | |

|  |
| --- |
| Orders |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Number | order\_date | cust\_id | salesperson\_id | Amount | | 10 | 8/2/96 | 4 | 2 | 540 | | 20 | 1/30/99 | 4 | 8 | 1800 | | 30 | 7/14/95 | 9 | 1 | 460 | | 40 | 1/29/98 | 7 | 2 | 2400 | | 50 | 2/3/98 | 6 | 7 | 600 | | 60 | 3/2/98 | 6 | 7 | 720 | | 70 | 5/6/98 | 9 | 7 | 150 | |

1. The names of all salespeople that have an order with Samsonic.

SELECT DISTINCT t1.Name

FROM Salesperson t1

JOIN Orders t2

ON t1.ID = t2.salesperson\_id

JOIN Customer t3

ON t2.cust\_id = t3.ID

WHERE t3.Name = 'Samsonic';

1. The names of all salespeople that do not have any order with Samsonic.

SELECT t1.Name

FROM Salesperson t1

WHERE t1.ID NOT IN (

SELECT t3.salesperson\_id

FROM Orders t3

JOIN Customer t2

ON t3.cust\_id = t2.ID

WHERE t2.Name = 'Samsonic'

);

1. The names of salespeople that have 2 or more orders.

SELECT t1.Name

FROM Salesperson t1

JOIN Orders t3 ON t1.ID = t3.salesperson\_id

GROUP BY t1.ID, t1.Name

HAVING COUNT(\*) >= 2;

1. Write a SQL statement to insert rows into a table called highAchiever(Name, Age), where a salesperson must have a salary of $100,000 or greater to be included in the table.

CREATE TABLE highAchiever (

Name VARCHAR(100),

Age INT

);

INSERT INTO highAchiever(Name, Age)

SELECT Name, Age

FROM Salesperson

WHERE Annual\_Salary >= 100000;

1. Given the 2 tables below, User and UserHistory:

|  |
| --- |
| **User**  user\_id  name  phone\_num |
|  |
| **UserHistory** user\_id  date  action |

* 1. Write a SQL query that returns the name, phone number and most recent date for any user that has logged in over the last 30 days (you can tell a user has logged in if the action field in UserHistory is set to "logged\_on").

SELECT t1.name, t1.phone\_num, MAX(t2.date) AS recent\_login

FROM User t1

JOIN UserHistory t2

ON t1.user\_id = t2.user\_id

WHERE t2.action = 'logged\_on'

AND t2.date >= DATE\_SUB(CURDATE(), INTERVAL 30 DAY)

GROUP BY t1.user\_id, t1.name, t1.phone\_num;