**Anshu Tripathy  
2017-11-16**

**Interview Problems:**

**SQL**

1. Write a query to select all rows from person. If the person row has a value in preferred\_first\_name, select the preferred name instead of the value in first name. Alias the column as REPORTING\_NAME.

*Solution:*

**SELECT person\_id, first\_name as REPORTING\_NAME, last\_name, date\_of\_birth, hire\_date, occupation FROM person WHERE preferred\_first\_name is null or preferred\_first\_name=’’ UNION SELECT person\_id, preferred\_first\_name as REPORTING\_NAME, last\_name, date\_of\_birth, hire\_date, occupation FROM person WHERE preferred\_first\_name <> ‘’;**

Output: In person table there is a record with preferred\_first\_name as “Anshu” and first\_name as “Anshuman”. There is another record with **null** preferred\_first\_name and first\_name as “Sriprasad”. This is the Output on running the Query-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PERSON\_ID | REPORTING\_NAME | LAST\_NAME | DATE\_OF\_BIRTH | HIRE\_DATE | OCCUPATON |
| 1 | ANSHU | TRIPATHY | 1989-08-11 | 2017-11-27 | ENGINEER |
| 2 | SRIPRASAD | KANNAN | 1988-06-25 | 2016-12-22 | ENGINEER |

1. Write a query to select all rows from person that have a NULL occupation.

*Solution:*

**SELECT \* FROM person WHERE occupation is null;**

Output: In person table there are two records that have null in their occupation column. This is the Output on running the Query-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PERSON\_ID | REPORTING\_NAME | LAST\_NAME | DATE\_OF\_BIRTH | HIRE\_DATE | OCCUPATON |
| 3 | RAJIV | MASAND | 1977-04-14 | 2017-08-30 | NULL |
| 4 | JOHN | DOE | 1963-01-21 | 2015-07-01 | NULL |

1. Write a query to select all rows from person that have a date\_of\_birth before August 7th, 1990.

*Solution:*

**SELECT \* FROM person WHERE date\_of\_birth < '1990-08-07';**

Output: In the person table there are four records that have values in date\_of\_birth column lesser than August 7th 1990. This is the output on running the Query-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PERSON\_ID | REPORTING\_NAME | LAST\_NAME | DATE\_OF\_BIRTH | HIRE\_DATE | OCCUPATON |
| 1 | ANSHU | TRIPATHY | 1989-08-11 | 2017-11-27 | ENGINEER |
| 2 | SRIPRASAD | KANNAN | 1988-06-25 | 2016-12-22 | ENGINEER |
| 3 | RAJIV | MASAND | 1977-04-14 | 2017-08-30 | NULL |
| 4 | JOHN | DOE | 1963-01-21 | 2015-07-01 | NULL |

1. Write a query to select all rows from person that have a hire\_date in the past 100 days.

*Solution:*

**select \* FROM person WHERE hire\_date > date\_sub(current\_date, interval 100 day);**

Output: In the person table there are two records where the hire date is in the last 100 days. This is the Output on running the Query-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PERSON\_ID | REPORTING\_NAME | LAST\_NAME | DATE\_OF\_BIRTH | HIRE\_DATE | OCCUPATON |
| 1 | ANSHU | TRIPATHY | 1989-08-11 | 2017-11-27 | ENGINEER |
| 3 | RAJIV | MASAND | 1977-04-14 | 2017-08-30 | NULL |

1. Write a query to select rows from person that also have a row in address with address\_type = “HOME”.

*Solution:*

**SELECT \* FROM person WHERE person\_id in (SELECT person\_id FROM address WHERE address\_type = "Home");**

Output: There are two records in person table that have their corresponding address\_type values as “HOME” in address table. This is the output on running the Query-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PERSON\_ID | REPORTING\_NAME | LAST\_NAME | DATE\_OF\_BIRTH | HIRE\_DATE | OCCUPATON |
| 1 | ANSHU | TRIPATHY | 1989-08-11 | 2017-11-27 | ENGINEER |
| 2 | SRIPRASAD | KANNAN | 1988-06-25 | 2016-12-22 | ENGINEER |

1. Write a query to select all rows from person and only those rows from address that have a matching billing address(address\_type = ‘BILL’). If a matching billing address does not exist, display ‘NONE’ in the address\_type column.

*Solution:*

**SELECT p.\*, a.\* FROM person p LEFT JOIN address a ON p.person\_id=a.person\_id WHERE a.address\_type='BILL';**

Output: There are two records in person table that have a record each in address table where the address\_type is ‘BILL’. This is the Output on running the Query-

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PERSON\_ID | REPORTING\_NAME | LAST\_NAME | DATE\_OF\_BIRTH | HIRE\_DATE | OCCUPATON | ADDRESS\_ID | PERSON\_ID | ADDRESS\_TYPE | STREET\_LINE\_1 | CITY | STATE | ZIP\_CODE |
| 1 | ANSHU | TRIPATHY | 1989-08-11 | 2017-11-27 | ENGINEER | 4 | 1 | BILL | 25 HIGHROSE STREET | NATICK | MA | 22345 |
| 4 | JOHN | DOE | 1963-01-21 | 2015-07-01 | NULL | 6 | 4 | BILL | 390 WESTROAD AVE. | FREMONT | CA | 60045 |

1. Write a query to count the number of addresses per address type.

*Solution:*

**SELECT address\_type, count(address\_id) as count FROM address group BY address\_type;**

Output: There are two records in address table that have their address\_type as ‘HOME’ and another two that have it as ‘BILL’. This is the Output on running the Query-

|  |  |
| --- | --- |
| ADDRESS\_TYPE | COUNT |
| HOME | 2 |
| BILL | 2 |

1. Write a query to select data in the following format:

|  |  |  |
| --- | --- | --- |
| LAST\_NAME | HOME\_ADDRESS | BILLING\_ADDRESS |
| SMITH | 89 Lyon Circle, Clifton, VA 12345 | 25 Science Park, New Haven, CT 06511 |
| JONES | 212 Maple Ave, Manassas, VA 22033 | 275 Winchester Ave, New Haven, CT 06511 |

*Solution:*

**SELECT t1.last\_name, t1.Home as 'home\_address', t2.Bill as 'billing\_address'**

**FROM (select p.person\_id, p.last\_name, CONCAT(a.street\_line\_1,' ,', a.city,' ,',a.state,' ,',a.zip\_code) AS 'Home' FROM person p join address a**

**ON**

**p.person\_id = a.person\_id**

**WHERE p.person\_id IN (SELECT person\_id FROM address WHERE address\_type = "BILL" AND p.person\_id IN (SELECT person\_id FROM address WHERE address\_type = "Home")) AND a.address\_type = "Home") as t1**

**JOIN**

**(SELECT p.person\_id, p.last\_name, CONCAT(a.street\_line\_1,' ,', a.city,' ,',a.state,' ,',a.zip\_code) AS 'BILL' FROM person p JOIN address a ON**

**p.person\_id = a.person\_id**

**WHERE p.person\_id IN (SELECT person\_id FROM address WHERE address\_type = "BILL" AND p.person\_id IN (SELECT person\_id FROM address WHERE address\_type = "Home")) AND a.address\_type = "BILL") AS t2**

**ON t1.person\_id = t2.person\_id;**

Output: There are two records in person table that each have two records in address table with one of each address\_type i.e. ‘HOME’ and ‘BILL’. This is the Output on running the Query-

|  |  |  |
| --- | --- | --- |
| LAST\_NAME | HOME\_ADDRESS | BILLING\_ADDRESS |
| TRIPATHY | 12 Peachwood Rd, Dunwoody, Atlanta, GA-42295 | 25 HighRose Street, Natick, MA-22345 |
| DOE | 220 ElmWood Ave., Sunnyvale, CA-60023 | 390 WestRoad Ave., Fremont, CA-60045 |

1. Write a query to update the *person.occupation* column to ‘X’ for all rows that have a ‘BILL’ address in the address table.

*Solution:*

**UPDATE person SET occupation = "X" WHERE person\_id IN (SELECT person\_id FROM address WHERE address\_type = "BILL");**

Output: There are two records in table person whose corresponding address\_type is ‘BILL’ in address table. This is the Output on running the Query-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PERSON\_ID | REPORTING\_NAME | LAST\_NAME | DATE\_OF\_BIRTH | HIRE\_DATE | OCCUPATON |
| 1 | ANSHU | TRIPATHY | 1989-08-11 | 2017-11-27 | X |
| 4 | JOHN | DOE | 1963-01-21 | 2015-07-01 | X |

PROBLEM

1. Given any three words for example LATE, SPUD, TILT, using one letter from each return all of the unique letter combinations.

*Solution:*

Declare Function for returning unique letter combinations that takes three strings as parameters and has return type of ArrayList of String {

declare char[] a by doing a toCharArray of first String;  
 declare char[] b by doing a toCharArray of second String;  
 declare char[] c by doing a toCharArray of third String;  
   
 declare ArrayList<String> combinations;

declare Object of Random Class rand;

Iterate through all three char arrays {

Extract a random character from char array ‘a’ without repetition using rand;  
 Extract a random character from char array ‘b’ without repetition using rand;  
 Extract a random character from char array ‘c’ without repetition using rand;  
  
 Concatenate the three random letters and add it as a String to combinations;

}

//The arrayList combinations will now contain all unique letter combinations from //the three Strings which were passed in.

return combinations;

}

1. How could you adjust your solution to accommodate any number of words of random length.

*Solution:*

I could make use of Dynamic programming instead of looping through all the words at the same time.