

SALESFORCE PRODIGIES

SOQL Query Examples in Salesforce

1. Fetch the child record from parent record (Contact is child and Account is Parent) – Standard Objects

```
SELECT Id, Name, (SELECT Id, LastName FROM Contacts) FROM Account.
```

Note- Contacts is a **Child Relationship Name** which is lookup field in child object

2. Fetch the child record from parent record (Student__c is child and Teacher__c is Parent) – Custom Objects

```
SELECT Id, Name, (SELECT Id, Name FROM Students__r) FROM Teacher__c
```

Note - Students is a **Child Relationship Name**(appended ' __r' in Students) which is lookup field in child object

3. Fetch the Parent record from Child Record (Contact is child and Account is Parent) – Standard Object

```
SELECT Id, Account.Name FROM Contact
```

4. Fetch the Parent record from Child Record (Student__c is child and Teacher__c is Parent) – Custom Object

```
SELECT Id, Teacher__r.Name FROM Student__c
```

Note- Here we don't need to add **s** in Relationship

5. Fetch the Account that has no contact record

```
SELECT Id, Name FROM Account WHERE Id NOT IN (SELECT AccountId FROM Contact)
```

Note- AccountId(Id of Account that is associated with contact) is lookup field

6. Fetch the Latest Account Record

```
SELECT Id, Name, CreatedDate FROM Account ORDER BY CreatedDate DESC
```

7. Fetch the Account record which is group by Name

```
SELECT Count(Id), Name FROM Account GROUP BY Name
```

Note- We can not use count with the field that is used in Group By.
For example we can not count name because we are using Name field in Group By

8. Determine how many leads are associated with each LeadSource value

```
SELECT LeadSource, COUNT(Name) FROM Lead GROUP BY LeadSource
```

9. Fetch the Lead Record that are associate with each LeadSource that generated more than 10 times

```
SELECT LeadSource, COUNT(Name) FROM Lead GROUP BY LeadSource Having Count(Name)>10
```

10. Fetch the Lead record where name end with 'abc'

```
SELECT LeadSource, Name FROM Lead WHERE Name LIKE '%abc'
```

11. How to fetch all fields in SOQL Query?

Ans : Suppose you want to fetch all fields of **contact** object then you can use **FIELDS(ALL)**.

FIELDS(ALL) – This fetches all the fields of an object. This is similar like Select * from SQL.

SELECT FIELDS(ALL) FROM Contact Limit 10 (For Standard Object)

FIELDS(STANDARD) – This fetches all standard fields of an object.

SELECT FIELDS(STANDARD) FROM Contact Limit 10

FIELDS(CUSTOM) – This fetches all custom fields of an object.

SELECT FIELDS(CUSTOM) FROM Contact Limit 10

12. Fetch the records from the recycle bin using soql ?

If you try that in the developer console, though, you'll get an **“Unknown error parsing query”** message.

So, to execute the query, you need to open the anonymous code window from the developer console.

```
List<Account> acc = [SELECT Id, isDeleted FROM Account WHERE isDeleted = TRUE  
ALL ROWS];
```

```
system.debug('Deleted Account>>>' + acc);
```

13. For Update Clause in soql ?

FOR UPDATE to lock sObject records while they're being updated in order to prevent race

conditions and other thread safety problems.

While an sObject record is locked, no other client or user is allowed to make updates either through code or the Salesforce user interface. The client locking the records can perform logic on the records and make updates with the guarantee that the locked records won't be changed by another client during the lock period.

SELECT Id, Name FROM Account FOR UPDATE

14. Different Operators in SOQL ?

AND - Use AND to return records that meet two conditions.

This query returns all records that have the first name **Nripesh** and the last name kum.

SELECT Name FROM Contact WHERE FirstName = 'Nripesh' AND LastName= 'Kum'

OR - Use **OR** to return records that meet one of two conditions. This query returns records with the last name Nripesh or the last name Kumar.

SELECT Name, Email FROM Contact WHERE FirstName = 'Nripesh' OR LastName = 'Kumar'

IN- Use IN to return records that meet at least one of three or more conditions. The IN clause is commonly used to return the values of a picklist, or values from a LIST or SET. This query returns all records that have the last name **James**, **Barr**, **Nedaerk**, or **Forbes**.

SELECT Name, Email FROM Contact WHERE LastName IN ('James', 'Barr', 'Nedaerk', 'Forbes')

ASC - Returns results in ascending order

SELECT Name, Email FROM Contact ORDER BY Name ASC LIMIT 5

DESC- Returns results in descending order

SELECT Name FROM Contact ORDER BY Name DESC LIMIT 5

NULLS FIRST | LAST - Returns null records at the beginning (NULLS FIRST) or end (NULLS LAST)

SELECT Name, Email FROM Contact ORDER BY Email NULLS LAST

SELECT Name, Email FROM Contact ORDER BY Email NULLS First

15. We want only accounts that have a related contact with the last name

Forbes ?

SELECT Name, (SELECT Name FROM Contacts) FROM Account WHERE Id IN (SELECT AccountId FROM Contact WHERE LastName = 'Kumar')

Some Clauses in SOQL Query

In the above Queries we have used many clauses.

*** **NOT** - NOT keyword is used for negation.

*** **Group By** - GROUP BY is used with Aggregate functions to group the result set by single or multiple columns.

*** **Order By** - ORDER BY is used to sort the records in ascending(ASC) or descending(DESC) order. It is used after the WHERE clause.

*** **Like** - LIKE keyword allows selective queries using wildcards.

*** **Having** - HAVING is an optional clause that can be used in a SOQL query to filter results that aggregate functions return.

*** **WHERE** - We can add the condition with the help of WHERE Clause

Governor Limit for SOQL

Description	Synchronous Limit	Asynchronous Limit
Total number of SOQL queries issued ¹	100	200
Total number of records retrieved by SOQL queries	50,000	50,000