

**DHANANJAY
SALESFORCE
DEVELOPER**

LEARN BETTER & FASTER

Presentation by Dhananjay Aher



Today's Presentation

DISCUSSION POINTS

Introduction To Basic SOQL

DHANANJAY AHER

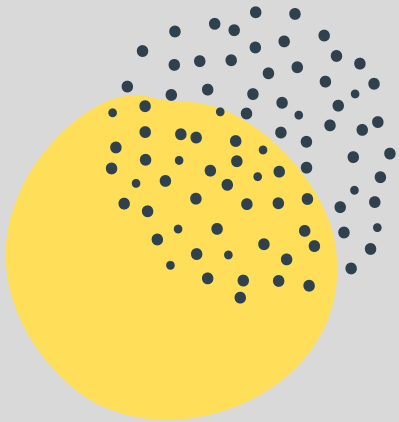


SOQL

Syntax

```
SELECT FIELD_NAME FROM OBJECT_NAME)
```

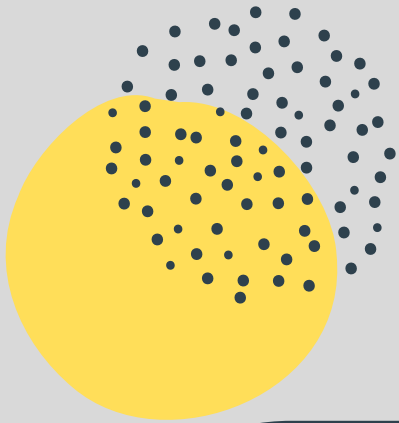
1.Fetch the child Record from Parent Record (Contact is child & Account is Parent) - Standard Object



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

```
SELECT Id,Name,(SELECT Id,Lastname FROM Contacts) FROM Account
```

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



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

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

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



Note– While fetching Parent from child – append . operator

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

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



Note– No need to add s in the Relationship

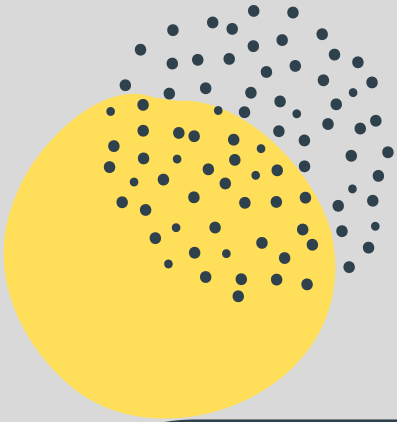
```
SELECT Id, College__r.Name FROM Student__c
```

**Fetch the Account which is not having any
contact associated with it**

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


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

Fetch the Latest Account Record



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

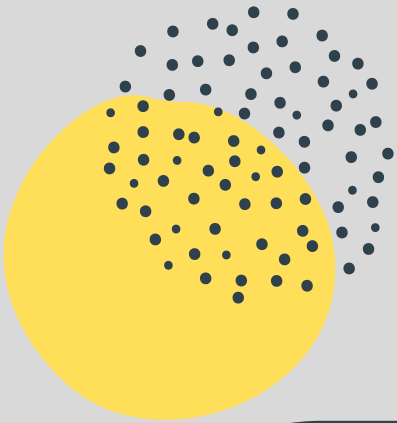
Fetch the Account record which is group by Name



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 B

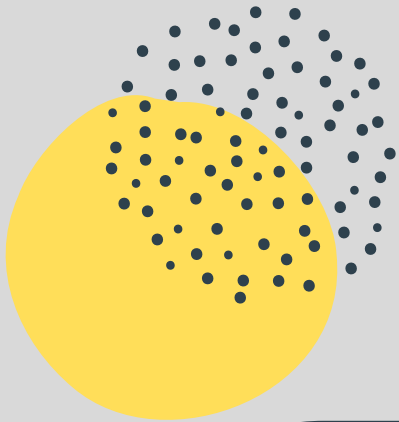
```
SELECT Count(Id), Name FROM Account GROUP BY Name
```

.Determine how many leads are associated with each LeadSource value



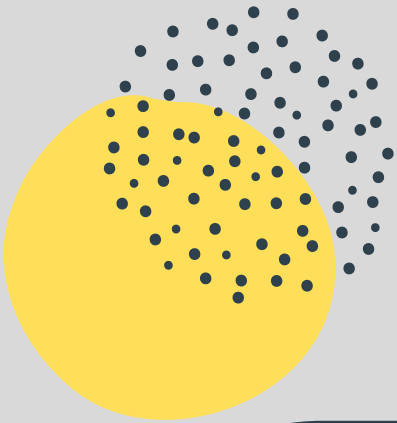
```
SELECT LeadSource, COUNT(Name) FROM Lead GROUP BY LeadSource
```

Fetch the Lead Record that are associate with each LeadSource that generated more than 20 times



```
SELECT LeadSource, COUNT(Name) FROM Lead GROUP BY LeadSource  
Having Count(Name)>20
```

Fetch the Lead record where name end with 'abc'



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

How to fetch all fields in SOQL Query



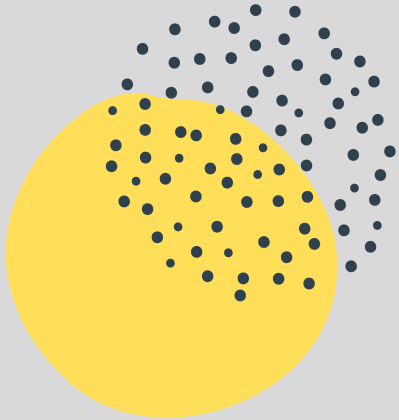
Note– The SOQL FIELDS function must have a LIMIT of at most 200

SELECT FIELDS(ALL) FROM Contact LIMIT 200

FIELDS(STANDARD) – This fetches all standard fields of an object.
SELECT FIELDS(STANDARD) FROM Contact Limit 10

SELECT FIELDS(CUSTOM) FROM Contact Limit 10

Fetch the records from the recycle bin using soql ?

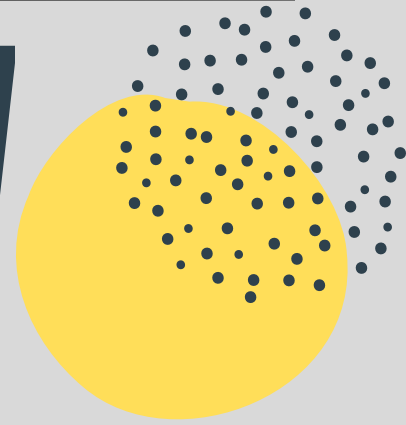


Note– If you try that in the developer console, though, you'll get an “Unknown error parsing query” message. So, to execute the query, you need to open the anonymous code window from the developer console

```
List acc = [SELECT Id, isDeleted FROM Account WHERE isDeleted = TRUE ALL ROWS];  
system.debug('Deleted Account>>>>'+acc)
```

For Update Clause in soql ?

Note- 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
```


Different Operators in SOQL ?

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

SELECT Name FROM Contact WHERE FirstName = 'abc' AND LastName= 'xyz'

OR - Use OR to return records that meet one of two conditions.

```
SELECT Name FROM Contact WHERE FirstName = 'abc' OR LastName= 'xyz'
```

IN- Use IN to return records that meet at least one of three or more conditions.

```
SELECT Name, Email FROM Contact WHERE LastName IN ('abc', 'xyz', 'mnq', 'pqr')
```

Different Operators in SQL ?

ASC - Returns results in ascending order

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

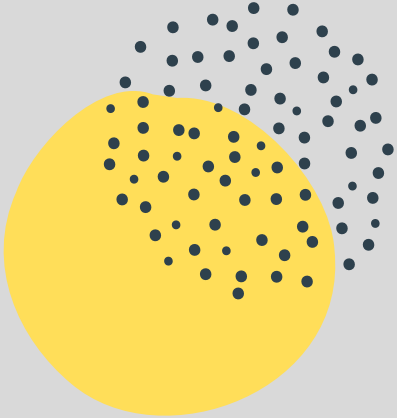
DESC- Returns results in descending order

```
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
```

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')
```

GROUP BY ROLLUP

Use the **GROUP BY ROLLUP** optional clause in a SQL query to add subtotals for aggregated data in query results. This action enables the query to calculate subtotals so that you don't have to maintain that logic in your code. You can use **GROUP BY ROLLUP** with aggregate functions, such as **SUM()** and **COUNT(fieldName)**.



```
[GROUP BY ROLLUP (fieldName[,...])]
```

GROUP BY CUBE

Use the GROUP BY CUBE clause in a SQL query to add subtotals for all combinations of a grouped field in the query results. This action is useful for compiling cross-tabular reports of data



```
[GROUP BY CUBE (fieldName[,...])]
```

HAVING

HAVING is an optional clause that can be used in a SQL query to filter results that aggregate functions return.



```
SELECT LeadSource, COUNT(Name) FROM Lead GROUP BY  
LeadSource HAVING COUNT(Name) > 100
```

OFFSET

When expecting many records in a query's results, you can display the results in multiple pages by using the OFFSET clause on a SQL query.

For example, you can use OFFSET to display records 51-75 and then jump to displaying records 301-350. Using OFFSET is an efficient way to handle large results sets.



```
SELECT Name FROM Merchandise__c WHERE Price__c > 5.0  
ORDER BY Name LIMIT 100 OFFSET 10
```

Thank
you!



DHANANJAY AHER

<https://www.linkedin.com/in/dhananjay-aher-343204192>