

CRM Practical 6

Name. Harsha Hargunani

Batch. E1

Roll no. 09

Username. harsha@bts.com

Aim - Illustrate the practical application of Data Manipulation Language (DML) statements, including Insert, Update, Delete, and Undelete operations, to navigate and manipulate data relationships within Salesforce, showcasing both Parent-to-Child and Child-to-Parent traversals.

INSERT

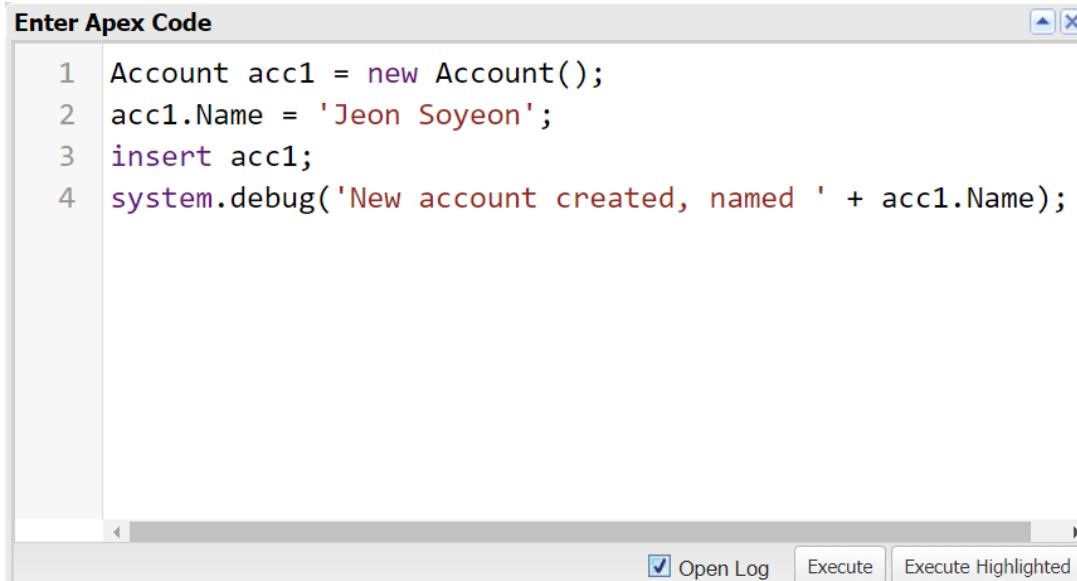
- In insert operation it is compulsory to insert All fields which are compulsory at metadata level and Validation Rule.
- While Inserting Fields we need to use the field/Api name.
- Salesforce recommends that DML statements Insert/Update/Delete should not be written in a Loop because if the loop runs more than 150 times, because DML Statement has a limit of 150 Records per transaction.

Q1. Insert a new record in standard object - Account.

Ans. Developer Console | Debug | Open Execute Anonymous Window

Code1:

```
Account acc1 = new Account();
acc1.Name = 'Jeon Soyeon';
insert acc1;
system.debug('New account created, named ' + acc1.Name);
```



The screenshot shows the 'Enter Apex Code' window in the Salesforce Developer Console. The code area contains the following Apex code:

```
1 Account acc1 = new Account();
2 acc1.Name = 'Jeon Soyeon';
3 insert acc1;
4 system.debug('New account created, named ' + acc1.Name);
```

At the bottom of the window, there are three buttons: 'Open Log' (with a checked checkbox), 'Execute', and 'Execute Highlighted'.

Below under the Logs section, check the "Debug Only" option.

| User | Application | Operation |
|------------------|-------------|--|
| Harsha Hargunani | Unknown | /services/data/v58.0/tooling/executeA... |

Output on Execute:

Log executeAnonymous @10/8/2023, 4:24:22 PM

| Timestamp | Event | Details |
|--------------|------------|--|
| 16:24:22:217 | USER_DEBUG | [4] DEBUG New account created, named Jeon Soyeon |

| Account Name | Account Site | Billing State/... | Phone | Type | Account Owner |
|---------------------------------|--------------|-------------------|----------------|--------------------|---------------|
| Sample Account for Entitlements | | | (014) 427-4427 | Customer - Channel | autproc |
| Pyramid Construction Inc. | | | | | HHarg |
| Ramdeobaba | | | | | HHarg |
| Jeon Soyeon | | | | | HHarg |
| University of Arizona | AZ | | (520) 773-9050 | Customer - Direct | HHarg |
| GenePoint | CA | | (650) 867-3450 | Customer - Channel | HHarg |
| sForce | CA | | (415) 901-7000 | | HHarg |
| Grand Hotels & Resorts Ltd | IL | | (312) 596-1000 | Customer - Direct | HHarg |

Code2:

```

Account acc1 = new Account();
acc1.Name = 'Jeon Jungkook';
acc1.Type = 'HYBE Labels Idol';
acc1.SLA_c = 'Silver';
acc1.Phone = '8888163473';
acc1.Ownership = 'Private';
acc1.Email_ID_c = 'jjk@hybelabels.com';
acc1.Industry = 'Idol';
insert acc1;
system.debug('New account created, named ' + acc1.Name);
system.debug('with email id ' + acc1.Email_ID_c);

```

Enter Apex Code

```

1 Account acc1 = new Account();
2 acc1.Name = 'Jeon Jungkook';
3 acc1.Type = 'HYBE Labels Idol';
4 acc1.SLA__c = 'Silver';
5 acc1.Phone = '8888163473';
6 acc1.Ownership = 'Private';
7 acc1.Email_ID__c = 'jjk@hybelabels.com';
8 acc1.Industry = 'Idol';
9 insert acc1;
10 system.debug('New account created, named ' + acc1.Name);
11 system.debug('with email id ' + acc1.Email_ID__c);

```

Open Log

Output on Execute:

Log executeAnonymous @10/8/2023, 4:35:07 PM

| Execution Log | | |
|---------------|------------|---|
| Timestamp | Event | Details |
| 16:35:07:163 | USER_DEBUG | [10] DEBUG New account created, named Jeon Jungkook |
| 16:35:07:163 | USER_DEBUG | [11] DEBUG with email id jjk@hybelabels.com |

Account | Salesforce **All Accounts | Accounts | Salesfo** **+**

16 items • Sorted by Account Name • Filtered by All accounts • Updated a few seconds ago

| | Account Name ↑ | Phone | Type | Ownership | SLA | Email ID | Account Owner Alias |
|----|-----------------------------------|----------------|--------------------|-----------|----------|--------------------|---------------------|
| 1 | Burlington Textiles Corp of Am... | (336) 222-7000 | Customer - Direct | Public | Silver | jjk@hybelabels.com | HHarg |
| 2 | Dickenson plc | (785) 241-6200 | Customer - Channel | Private | Bronze | | HHarg |
| 3 | Edge Communications | (512) 757-6000 | Customer - Direct | Public | Silver | | HHarg |
| 4 | Express Logistics and Transport | (503) 421-7800 | Customer - Channel | Public | Platinum | | HHarg |
| 5 | GenePoint | (650) 867-3450 | Customer - Channel | Private | Bronze | | HHarg |
| 6 | Grand Hotels & Resorts Ltd | (312) 596-1000 | Customer - Direct | Public | Gold | | HHarg |
| 7 | Jeon Jungkook | 8888163473 | HYBE Labels Idol | Private | Silver | jjk@hybelabels.com | HHarg |
| 8 | Jeon Soyeon | | | | | | HHarg |
| 9 | Pyramid Construction Inc. | (014) 427-4427 | Customer - Channel | Public | Silver | | HHarg |
| 10 | Ramdeobaba | | | | | | HHarg |
| 11 | Sample Account for Entitlements | | | | | | autproc |

35°C Partly sunny 04:37 PM 08-10-2023

Q2. Insert a new record in custom object - Guest along with DOB field.

Ans. Developer Console | Debug | Open Execute Anonymous Window

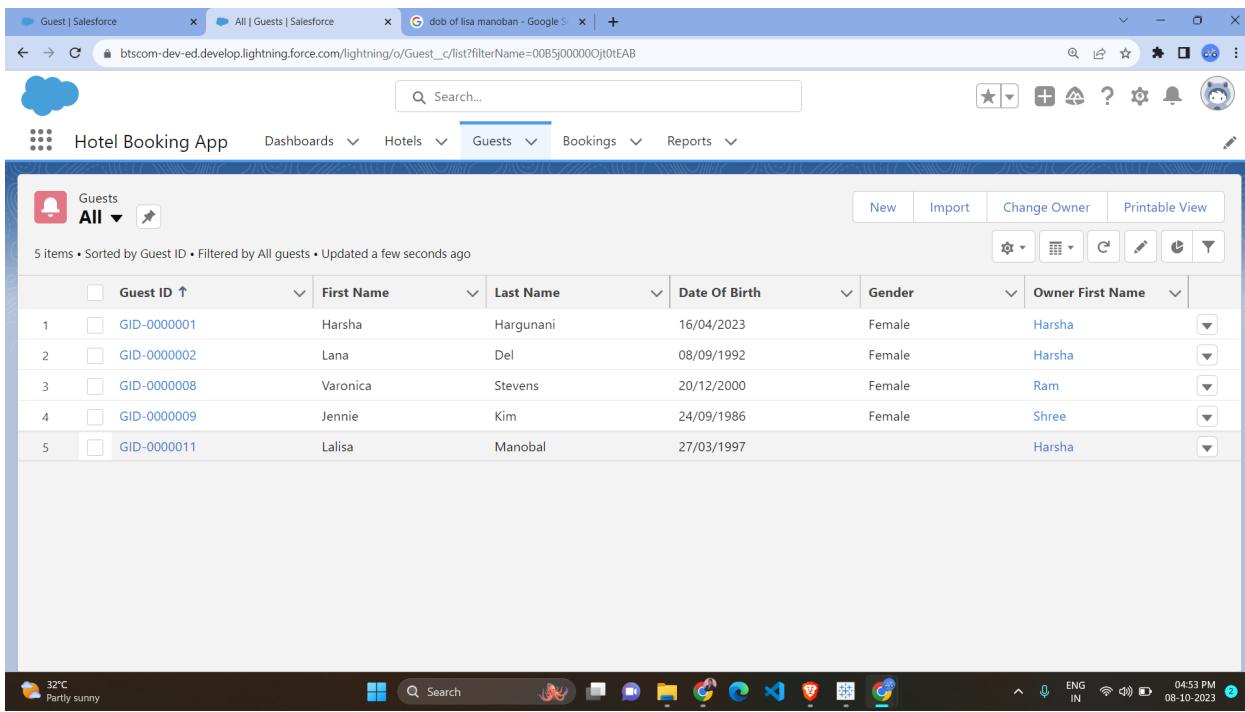
Code:



```
1 Guest__c g1 = new Guest__c();
2 g1.First_Name__c = 'Lalisa';
3 g1.Last_Name__c = 'Manobal';
4 g1.DOB__c = Date.parse('27/3/1997');
5 // date format - mm/dd/yyyy --> dd/mm/yyyy
6 insert g1;
7 system.debug('New Guest'+ g1.First_Name__c +'entry was successful');
```

The screenshot shows the 'Enter Apex Code' window in the Salesforce developer console. The code is written in Apex, creating a new Guest__c record named g1 with first name Lalisa and last name Manobal. The DOB is set to '27/3/1997'. A note in the code indicates that the date format 'mm/dd/yyyy' needs to be converted to 'dd/mm/yyyy'. The code concludes with a debug statement. At the bottom of the window are buttons for 'Open Log', 'Execute', and 'Execute Highlighted'.

Output:



The screenshot shows the 'Guests' list view in the Hotel Booking App. The interface includes a header with tabs for Guests, Bookings, and Reports. The Guests tab is selected. The list view displays 5 items, sorted by Guest ID. The columns include Guest ID, First Name, Last Name, Date Of Birth, Gender, and Owner First Name. The data shows five guest entries: Harsha Hargunani (DOB 16/04/2023), Lana Del (DOB 08/09/1992), Varonica Stevens (DOB 20/12/2000), Jennie Kim (DOB 24/09/1986), and Lalisa Manobal (DOB 27/03/1997). The owner for all guests is listed as Harsha. The bottom of the screen shows a taskbar with various icons and system status.

| | Guest ID ↑ | First Name | Last Name | Date Of Birth | Gender | Owner First Name |
|---|-------------|------------|-----------|---------------|--------|------------------|
| 1 | GID-0000001 | Harsha | Hargunani | 16/04/2023 | Female | Harsha |
| 2 | GID-0000002 | Lana | Del | 08/09/1992 | Female | Harsha |
| 3 | GID-0000008 | Varonica | Stevens | 20/12/2000 | Female | Ram |
| 4 | GID-0000009 | Jennie | Kim | 24/09/1986 | Female | Shree |
| 5 | GID-0000011 | Lalisa | Manobal | 27/03/1997 | | Harsha |

This can also be done by putting values in the constructor:

```
Guest__c g1 = new Guest__c(First_Name__c = 'Lalisa', DOB__c = Date.parse('27/3/1997'));
insert g1;
```

Q3. Perform an Insert operation to add 150 records to the Salesforce database.

Ans. Code:

```
For(Integer i=1; i<=150; i++) {  
    Account accObj = new Account(Name = 'Cinemax'+i);  
    insert accObj;  
}
```

Enter Apex Code

```
1 For(Integer i=1; i<=150; i++) {  
2     Account accObj = new Account(Name = 'Cinemax'+i);  
3     insert accObj;  
4 }
```

Open Log Execute Execute Highlighted

Output:

| | | |
|----|--------------------------|------------|
| 53 | <input type="checkbox"/> | Cinemax145 |
| 54 | <input type="checkbox"/> | Cinemax146 |
| 55 | <input type="checkbox"/> | Cinemax147 |
| 56 | <input type="checkbox"/> | Cinemax148 |
| 57 | <input type="checkbox"/> | Cinemax149 |
| 58 | <input type="checkbox"/> | Cinemax15 |
| 59 | <input type="checkbox"/> | Cinemax150 |

Q4. Compose an Apex code snippet demonstrating the addition of 200 records to the 'Account' object in Salesforce.

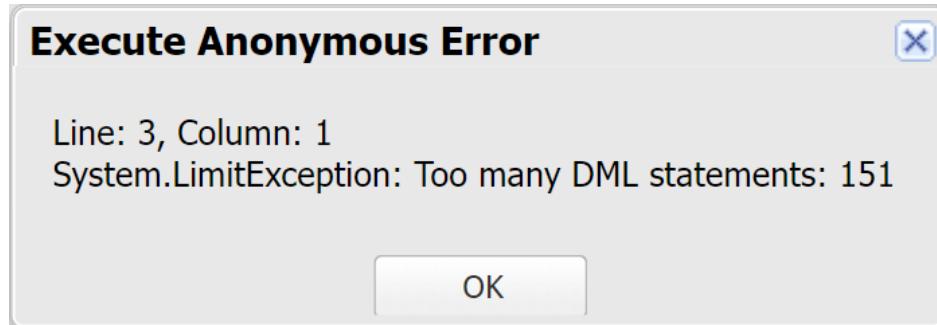
Ans. If i use the previous approach to code

Enter Apex Code

```
1 For(Integer i=1; i<=200; i++) {  
2     Account accObj = new Account(Name = 'MTV'+i);  
3     insert accObj;  
4 }
```

Open Log Execute Execute Highlighted

We get an error



Hence we follow the Bulkification approach. Bulkification means we can perform operations on multiple Records. Steps:

1. Create a List of Particular Data Type/Object.
2. Add Elements in the List.
3. Perform DML statements over the List.

Code:

```
list<Account> aList = new list<Account>();
for(integer i=0; i<=200; i++) {
    Account accObj = new Account();
    accObj.Name = 'MTV'+i;
    aList.add(accObj);
}
if(!aList.isEmpty()){
    insert aList;
}
```

The code editor window shows the same Apex code as above. At the bottom, there are three buttons: 'Open Log' (with a checked checkbox), 'Execute', and 'Execute Highlighted'.

```
1 list<Account> aList = new list<Account>();
2 for(integer i=0; i<=200; i++) {
3     Account accObj = new Account();
4     accObj.Name = 'MTV'+i;
5     aList.add(accObj);
6 }
7 if(!aList.isEmpty()){
8     insert aList;
9 }
```

Output:

Q5. Write an Apex code snippet that creates a list of Account records with different names and inserts them into Salesforce.

Ans. Code:

```
List<String> nameList = new List<String>();
nameList.Add('Bunty');
nameList.Add('Bubli');
nameList.Add('Rahul');
nameList.Add('Sonia');
nameList.Add('Munni');
nameList.Add('Amit');

List<Account> accountList=new List<Account>;
If(!nameList.isEmpty()){
    for(String name:nameList){
        Account objAccount = new Account();
        objAccount.Name = name;
        accountList.add(objAccount);
    }
    if(!accountList.isEmpty()){
        insert accountList;
    }
}
else{
```

```
System.debug('Name List Is Empty Add Account Names');
```

```
}
```

Q6. How to add a Child record and assign a Parent to it? Explain it by adding Contact Record and assigning Account Record For it.

Ans. Step1: Last Name is a mandatory field for creating a contact record.

Contact Information

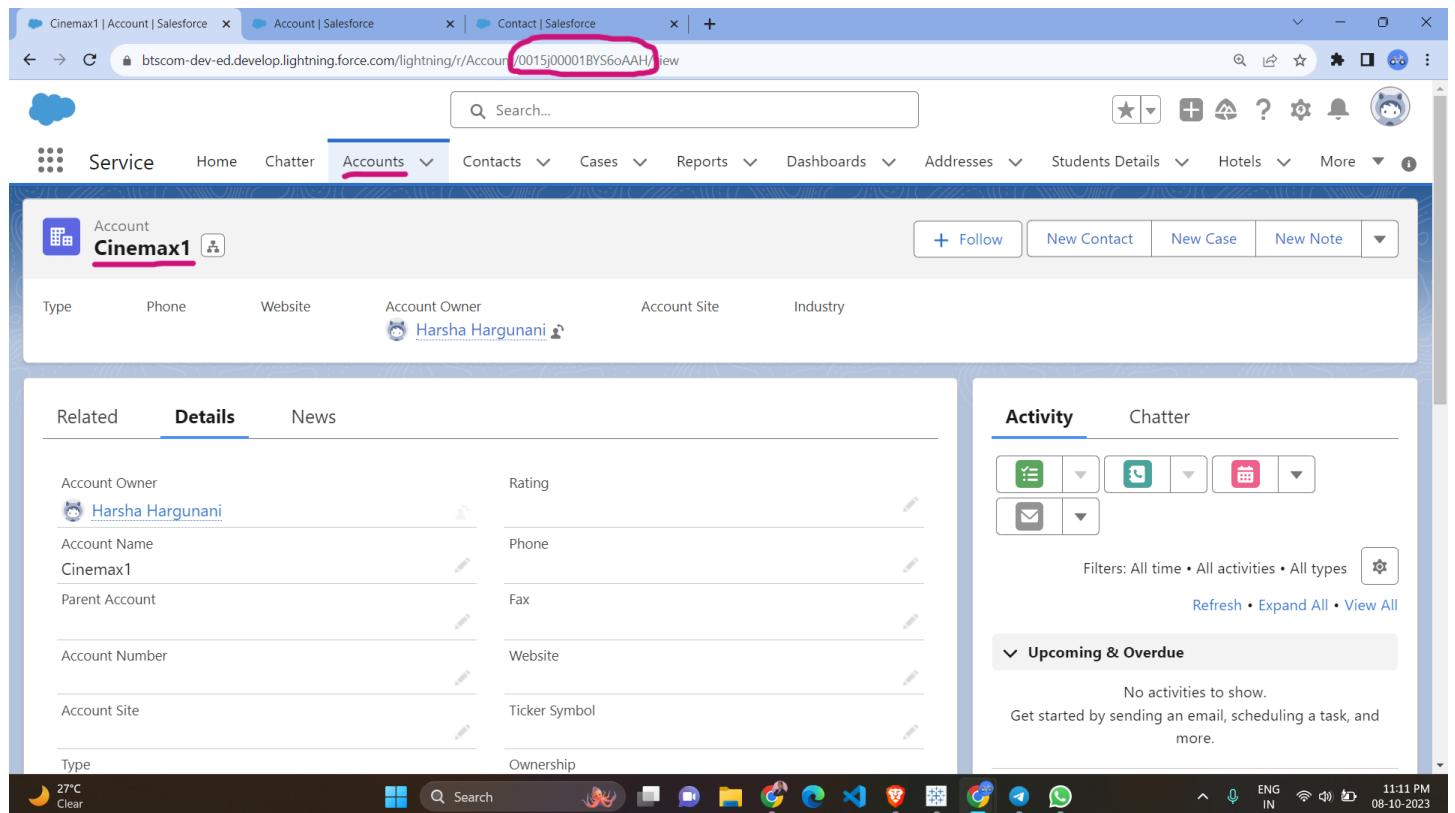
| | |
|---|---|
| Contact Owner  Harsha Hargunani | Phone <input type="text"/> |
| * Name Salutation <input type="text" value="--None--"/> | Home Phone <input type="text"/> |
| First Name <input type="text"/> | Mobile <input type="text"/> |
| * Last Name <input type="text"/> | Other Phone <input type="text"/> |
| Account Name <input type="text" value="Search Accounts..."/>  | Title <input type="text"/> |
| Department <input type="text"/> | <input type="button" value="Cancel"/> <input type="button" value="Save & New"/> <input type="button" value="Save"/> |

Step2: Field Name of the Lookup Relationship that connects Contact and Account Objects is "AccountId".

SETUP > OBJECT MANAGER
Contact

| Details | Fields & Relationships 36+ Items, Sorted by Field Label | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------------------------|---|-----------------|-------------------|-----------|-------------------|---------|--------------|-----------|-----------------|--|---|----------------|-------------------|----------------|--|---|--------------|-----------------|----------------|--|---|-----------|---------------|----------|--|--|-------------|----------------|-------|--|--|-----------|-----------|------|--|--|
| Fields & Relationships | <table border="1"><thead><tr><th>FIELD LABEL</th><th>FIELD NAME</th><th>DATA TYPE</th><th>CONTROLLING FIELD</th><th>INDEXED</th></tr></thead><tbody><tr><td>Account Name</td><td>AccountId</td><td>Lookup(Account)</td><td></td><td>✓</td></tr><tr><td>Account Rating</td><td>Account_Rating__c</td><td>Formula (Text)</td><td></td><td>▼</td></tr><tr><td>Account Type</td><td>Account_Type__c</td><td>Formula (Text)</td><td></td><td>▼</td></tr><tr><td>Assistant</td><td>AssistantName</td><td>Text(40)</td><td></td><td></td></tr><tr><td>Asst. Phone</td><td>AssistantPhone</td><td>Phone</td><td></td><td></td></tr><tr><td>Birthdate</td><td>Birthdate</td><td>Date</td><td></td><td></td></tr></tbody></table> | FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED | Account Name | AccountId | Lookup(Account) | | ✓ | Account Rating | Account_Rating__c | Formula (Text) | | ▼ | Account Type | Account_Type__c | Formula (Text) | | ▼ | Assistant | AssistantName | Text(40) | | | Asst. Phone | AssistantPhone | Phone | | | Birthdate | Birthdate | Date | | |
| FIELD LABEL | FIELD NAME | DATA TYPE | CONTROLLING FIELD | INDEXED | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Account Name | AccountId | Lookup(Account) | | ✓ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Account Rating | Account_Rating__c | Formula (Text) | | ▼ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Account Type | Account_Type__c | Formula (Text) | | ▼ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Assistant | AssistantName | Text(40) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Asst. Phone | AssistantPhone | Phone | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Birthdate | Birthdate | Date | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Page Layouts | <input type="button" value="New"/> <input type="button" value="Deleted Fields"/> <input type="button" value="Field Dependencies"/> <input type="button" value="Set History Tracking"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Lightning Record Pages | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Buttons, Links, and Actions | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Compact Layouts | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Field Sets | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Object Limits | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Record Types | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

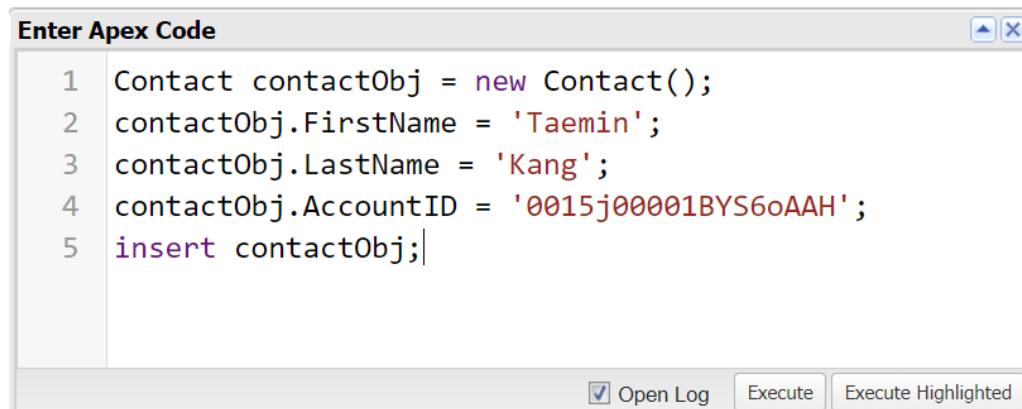
Step3: Copy the AccountID from the URL of the Account that'll be linked with Contact record later.



The screenshot shows the Salesforce Lightning interface. At the top, there are three tabs: 'Cinemax1 | Account | Salesforce', 'Account | Salesforce', and 'Contact | Salesforce'. The middle tab is active. Below the tabs, the URL in the browser bar is <https://btscom-ed-dev.lightning.force.com/lightning/r/Account/0015j00001BYS6oAAH/new>, with the Account ID '0015j00001BYS6oAAH' highlighted with a pink box. The main content area displays the 'Cinemax1' account details under the 'Details' tab. On the right side, there's an 'Activity' sidebar showing no upcoming activities. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 08-10-2023 at 11:11 PM.

Code:

```
Contact contactObj = new Contact();
contactObj.FirstName = 'Taemin';
contactObj.LastName = 'Kang';
contactObj.AccountID = '0015j00001BYS6oAAH';
insert contactObj;
```



The screenshot shows the 'Enter Apex Code' window. The code area contains the following Apex code:

```
1 Contact contactObj = new Contact();
2 contactObj.FirstName = 'Taemin';
3 contactObj.LastName = 'Kang';
4 contactObj.AccountID = '0015j00001BYS6oAAH';
5 insert contactObj;
```

At the bottom of the window, there are three buttons: 'Open Log' (with a checked checkbox), 'Execute', and 'Execute Highlighted'.

Output:

| | <input type="checkbox"/> Name | Account Name | Title ↑ | Phone | Email | Contact Status |
|---|--|------------------------|---------|----------------|---------------|----------------|
| 1 | <input type="checkbox"/> Siddartha Nedaerk | sForce | | | | HHarg |
| 2 | <input type="checkbox"/> Jake Llorac | sForce | | | | HHarg |
| 3 | <input type="checkbox"/> Taemin Kang | Cinemax1 | | | | HHarg |
| 4 | <input type="checkbox"/> Arthur Song | United Oil & Gas Corp. | CEO | (212) 842-5500 | asong@uog.com | HHarg |
| 5 | <input type="checkbox"/> Sean Forbes | Edge Communications | CFO | (512) 757-6000 | sean@edge.com | HHarg |

Q7. Provide an example in Apex of inserting a Contact record using Database statements.

Ans.

| Sr. no. | DML Statements | Database Methods |
|---------|---|---|
| 1. | Partial Update is not allowed. For example, if you have 100 records in a list, then either all the records will be updated or none. | Partial update is allowed. You can specify the Parameter in Database method as true or false, true to allow the partial update and false for not allowing the same. |
| 2. | You cannot get the list of success and failed records. | You can get the list of success and failed records as we have seen in the example. |
| 3. | ConvertLead Not Available in DML Statements | ConvertLead Operation is only available in Database class. |
| 4. | Example – insert accounts; | Example – Database.insert(listName, False), where false indicates that partial update is not allowed. |

Code:

```
Contact contactObj = new Contact();
contactObj.FirstName = 'Jihyo';
contactObj.LastName = 'Park';
contactObj.AccountID = '0015j00001BWKEeAAP';
Database.insert(contactObj, False);
```

Enter Apex Code

```
1 Contact contactObj = new Contact();
2 contactObj.FirstName = 'Jihyo';
3 contactObj.LastName = 'Park';
4 contactObj.AccountID = '0015j00001BWKEeAAP';
5 Database.insert(contactObj, False);
```

Open Log Execute Execute Highlighted

Output:

The screenshot shows the Salesforce 'Recently Viewed' contacts list. At the top, there are buttons for 'New', 'Import', 'Add to Campaign', and 'Send List Email'. Below that is a search bar with placeholder text 'Search this list...'. The main area displays a table with columns: Name, Account Name, Account..., Phone, Email, and Contact O... (with a dropdown arrow). There are two rows of data:

| | Name | Account Name | Account... | Phone | Email | Contact O... |
|---|-------------|--------------|------------|-------|-------|----------------------------------|
| 1 | Jihyo Park | Ramdeobaba | | | HHarg | <input type="button" value="▼"/> |
| 2 | Taemin Kang | Cinemax1 | | | HHarg | <input type="button" value="▼"/> |

Q8. Demonstrate difference between DML statement and Database method while inserting an incomplete record.

Ans. We know that Last Name is a mandatory field for the Contact object.

1. DML Statement:

Code:

```
1 Contact contactObj = new Contact();
2 contactObj.FirstName = 'Monika';
3 contactObj.AccountID = '0015j00001BWKEeAAP';
4 insert contactObj;
5 system.debug('New record was successfully created!');
```

The screenshot shows the 'Enter Apex Code' window. It contains the following Apex code:

```
1 Contact contactObj = new Contact();
2 contactObj.FirstName = 'Monika';
3 contactObj.AccountID = '0015j00001BWKEeAAP';
4 insert contactObj;
5 system.debug('New record was successfully created!');
```

At the bottom, there are buttons for 'Open Log', 'Execute', and 'Execute Highlighted'. The 'Execute' button is highlighted with a checked checkbox.

Output:

The screenshot shows the 'Execute Anonymous Error' window. It displays the following error message:

Line: 4, Column: 1
System.DmlException: Insert failed. First exception on row 0; first error: REQUIRED_FIELD_MISSING, Required fields are missing: [LastName]: [LastName]

At the bottom, there is an 'OK' button.

2. Database Methods

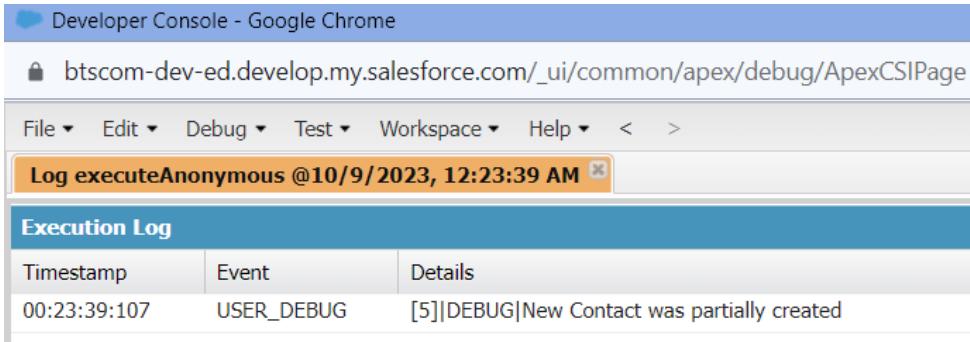
Code:



```
1 Contact contactObj = new Contact();
2 contactObj.FirstName = 'Nayeon';
3 contactObj.AccountID = '0015j00001BWKEeAAP';
4 Database.insert(contactObj, False);
5 system.debug('New Contact was partially created');
```

The screenshot shows the "Enter Apex Code" window from the Salesforce developer console. The code is displayed in a text area, and at the bottom, there are three buttons: "Open Log" (with a checked checkbox), "Execute", and "Execute Highlighted".

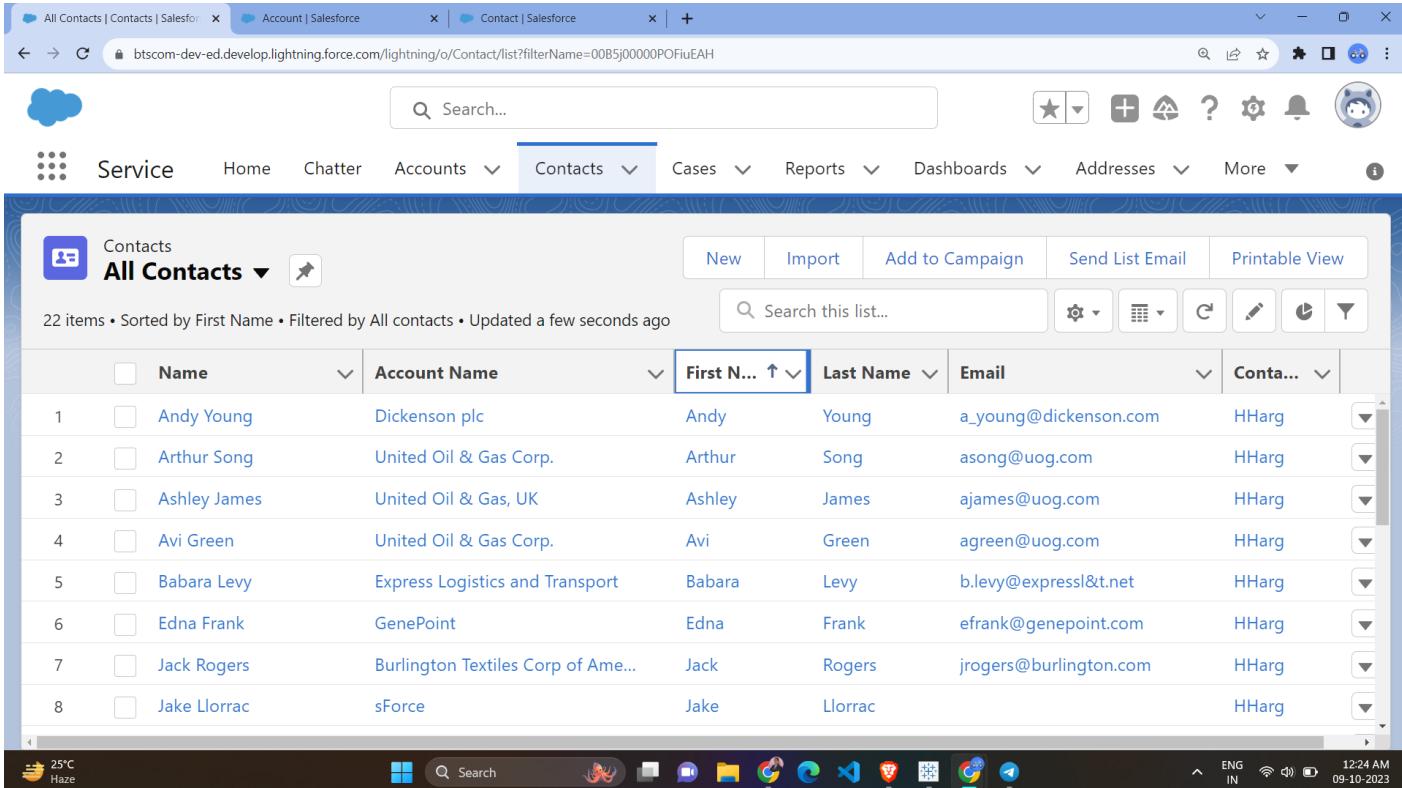
Output:



The screenshot shows the "Developer Console - Google Chrome" window. The URL is btscom-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage. The "Log executeAnonymous @10/9/2023, 12:23:39 AM" tab is selected. Below it, the "Execution Log" table shows a single entry:

| Timestamp | Event | Details |
|--------------|------------|--|
| 00:23:39:107 | USER_DEBUG | [5] DEBUG New Contact was partially created |

No error thrown, but no record was created either. Hence, Database methods can perform partial operations.



The screenshot shows the Salesforce Lightning interface. The top navigation bar includes "All Contacts", "Contacts", "Cases", "Reports", "Dashboards", "Addresses", "More", and a search bar. The main content area displays the "All Contacts" page with the following data:

| # | Name | Account Name | First Name | Last Name | Email | Cont... |
|---|--------------|------------------------------------|------------|-----------|-------------------------|---------|
| 1 | Andy Young | Dickenson plc | Andy | Young | a.young@dickenson.com | HHarg |
| 2 | Arthur Song | United Oil & Gas Corp. | Arthur | Song | asong@uog.com | HHarg |
| 3 | Ashley James | United Oil & Gas, UK | Ashley | James | ajames@uog.com | HHarg |
| 4 | Avi Green | United Oil & Gas Corp. | Avi | Green | agreen@uog.com | HHarg |
| 5 | Babara Levy | Express Logistics and Transport | Babara | Levy | b.levy@expressl&t.net | HHarg |
| 6 | Edna Frank | GenePoint | Edna | Frank | e.frank@gene-point.com | HHarg |
| 7 | Jack Rogers | Burlington Textiles Corp of Ame... | Jack | Rogers | j.rogers@burlington.com | HHarg |
| 8 | Jake Llorac | sForce | Jake | Llorac | | HHarg |

The page also includes a toolbar with "New", "Import", "Add to Campaign", "Send List Email", and "Printable View" buttons, along with a search bar and various filter and sort options.

If the boolean value is set to True, the database method will perform an All or Nothing operation.



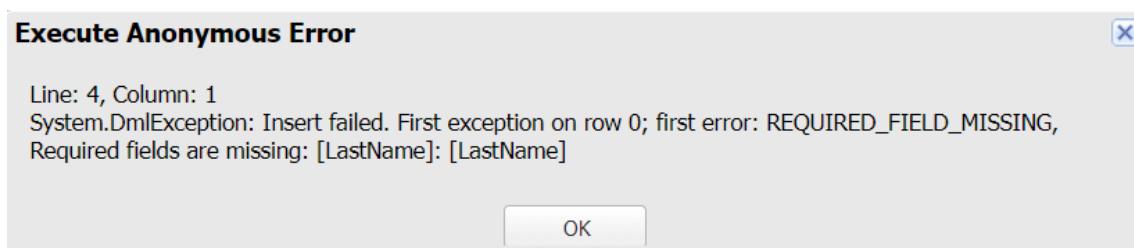
```
1 Contact contactObj = new Contact();
2 contactObj.FirstName = 'Nayeon';
3 contactObj.AccountID = '0015j00001BWKEeAAP';
4 Database.insert(contactObj, True);
5 system.debug('New Contact was partially created');
```

The screenshot shows the 'Enter Apex Code' window with the following code:

```
1 Contact contactObj = new Contact();
2 contactObj.FirstName = 'Nayeon';
3 contactObj.AccountID = '0015j00001BWKEeAAP';
4 Database.insert(contactObj, True);
5 system.debug('New Contact was partially created');
```

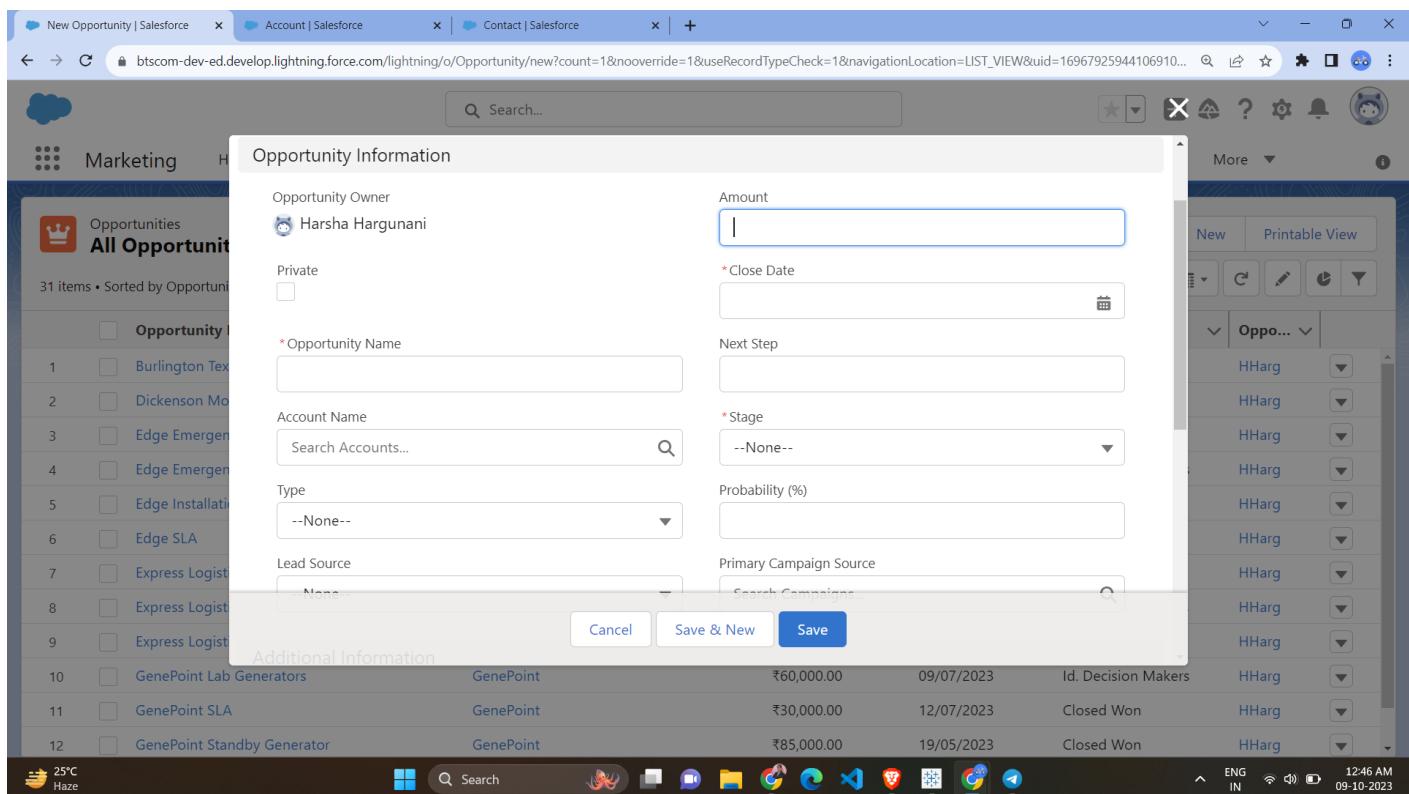
At the bottom, there are three buttons: 'Open Log' (checked), 'Execute', and 'Execute Highlighted'.

That is, we get the same error as in DML if an incomplete record is attempted to be created.



Q9. Write Apex code to insert 200 records for Opportunity Object.

Ans. List the mandatory fields for Opportunity Object.



The screenshot shows the 'Opportunity Information' page in the Salesforce Lightning interface. The mandatory fields listed are:

- Amount
- *Close Date
- *Opportunity Name
- Account Name
- *Stage
- Type
- Probability (%)
- Lead Source
- Primary Campaign Source

The 'Opportunity Owner' field is populated with 'Harsha Hargunani'. The 'Private' checkbox is unchecked. The 'Next Step' field is empty. The 'Stage' dropdown is set to '--None--'. The 'Probability (%)' field is empty. The 'Primary Campaign Source' field is empty. The 'Additional Information' section shows 12 rows of opportunity data, each with columns: Name, Type, Amount, Close Date, Stage, and Decision Makers.

1. Opportunity Name
2. Close Date
3. Stage

Code:

```
List<Opportunity> oppList=New List<Opportunity>();  
  
For(Integer i=1; i<=200; i++){  
    Opportunity objOpp = new Opportunity();  
    objOpp.Name = 'Golden opp'+i;  
    objOpp.CloseDate = Date.newInstance(2023, 08, 01);  
    objOpp.StageName = 'Qualification';  
  
    oppList.add(objOpp);  
}  
  
insert oppList;
```

Enter Apex Code

```
1  List<Opportunity> oppList=New List<Opportunity>();  
2  
3  For(Integer i=1; i<=200; i++){  
4      Opportunity objOpp = new Opportunity();  
5      objOpp.Name = 'Golden opp'+i;  
6      objOpp.CloseDate = Date.newInstance(2023, 08, 01);  
7      objOpp.StageName = 'Qualification';  
8  
9      oppList.add(objOpp);  
10 }  
11  
12 insert oppList;
```

Open Log Execute Execute Highlighted

Output:

All Opportunities | Opportunities | Account | Salesforce | Contact | Salesforce

Opportunities

Marketing Home Chatter Campaigns Leads Contacts Opportunities Reports Dashboards Addresses More

Opportunities All Opportunities

150+ items • Sorted by Opportunity Name • Filtered by All opportunities • Updated a few seconds ago

| Opportunity Name ↑ | Account Name | Amount | Close Date | Stage | Oppo... |
|--------------------|--------------|--------|------------|---------------|---------|
| 117 Golden opp193 | | | 01/08/2023 | Qualification | HHarg |
| 118 Golden opp194 | | | 01/08/2023 | Qualification | HHarg |
| 119 Golden opp195 | | | 01/08/2023 | Qualification | HHarg |
| 120 Golden opp196 | | | 01/08/2023 | Qualification | HHarg |
| 121 Golden opp197 | | | 01/08/2023 | Qualification | HHarg |
| 122 Golden opp198 | | | 01/08/2023 | Qualification | HHarg |
| 123 Golden opp199 | | | 01/08/2023 | Qualification | HHarg |
| 124 Golden opp2 | | | 01/08/2023 | Qualification | HHarg |
| 125 Golden opp20 | | | 01/08/2023 | Qualification | HHarg |
| 126 Golden opp200 | | | 01/08/2023 | Qualification | HHarg |
| 127 Golden opp21 | | | 01/08/2023 | Qualification | HHarg |
| 128 Golden opp22 | | | 01/08/2023 | Qualification | HHarg |

25°C Haze

Search

ENG IN 12:51 AM 09-10-2023

Q10. Demonstrate Database insertion method with one complete and one incomplete record.

Display the error message.

Ans.

Code:

```
List<Contact> conList = new List<Contact>();  
Contact objCon1 = new Contact();  
objCon1.FirstName = 'Minho';  
objCon1.LastName = 'Kim';  
Contact objCon2 = new Contact();  
objCon2.FirstName = 'Woobin';  
conList.add(objCon1);  
conList.add(objCon2);  
  
if(!conList.isEmpty()){  
    Database.SaveResult[] Result = Database.Insert(conList, False);  
    For(Database.SaveResult sr:result){  
        if(sr.isSuccess()){  
            system.debug('Successfully inserted Contact, contact ID: '+sr.getID());  
        }  
        Else{  
            For(Database.Error err:sr.getErrors()){  
                system.debug('The Following Errors Has Occured');  
            }  
        }  
    }  
}
```

```

        system.debug(err.getStatusCode()+' : '+err.getMessage());
        system.debug('Contact Fields that affected this Error: '+err.getFields());
    }
}
}
}

```

Enter Apex Code

```

1 List<Contact> conList = new List<Contact>();
2 Contact objCon1 = new Contact();
3 objCon1.FirstName = 'Minho';
4 objCon1.LastName = 'Kim';
5 Contact objCon2 = new Contact();
6 objCon2.FirstName = 'Woobin';
7 conList.add(objCon1);
8 conList.add(objCon2);
9
10 if(!conList.isEmpty()){
11     Database.SaveResult[] Result = Database.Insert(conList, False);
12     For(Database.SaveResult sr:result){
13         if(sr.isSuccess()){
14             system.debug('Successfully inserted Contact, contact ID: '+sr.getID());
15         }
16     Else{
17         For(Database.Error err:sr.getErrors()){
18             system.debug('The Following Errors Has Occured');
19             system.debug(err.getStatusCode()+' : '+err.getMessage());
20             system.debug('Contact Fields that affected this Error: '+err.getFields());
21         }
22     }
23 }
24

```

Open Log

Output:

Log executeAnonymous @10/9/2023, 1:07:53 AM

| Execution Log | | |
|---------------|------------|---|
| Timestamp | Event | Details |
| 01:07:53:168 | USER_DEBUG | [14] DEBUG Successfully inserted Contact, contact ID: 0035j00001BFIXWAA5 |
| 01:07:53:169 | USER_DEBUG | [18] DEBUG The Following Errors Has Occured |
| 01:07:53:185 | USER_DEBUG | [19] DEBUG REQUIRED_FIELD_MISSING:Required fields are missing: [LastName] |
| 01:07:53:185 | USER_DEBUG | [20] DEBUG Contact Fields that affected this Error: (LastName) |

| All Contacts | | | | | | | Actions | | |
|--------------|-------------|-------------------------------------|------------|-----------|------------------------|---------------------|---------|--|--|
| | Name | Account Name | First Name | Last Name | Email | Contact Owner Alias | | | |
| 1 | Minho Kim | | Minho | Kim | | HHarg | | | |
| 2 | Jack Rogers | Burlington Textiles Corp of America | Jack | Rogers | jrogers@burlington.com | HHarg | | | |
| 3 | Taemin Kang | Cinemax1 | Taemin | Kang | | HHarg | | | |
| 4 | Andy Young | Dickenson plc | Andy | Young | a_young@dickenson.com | HHarg | | | |

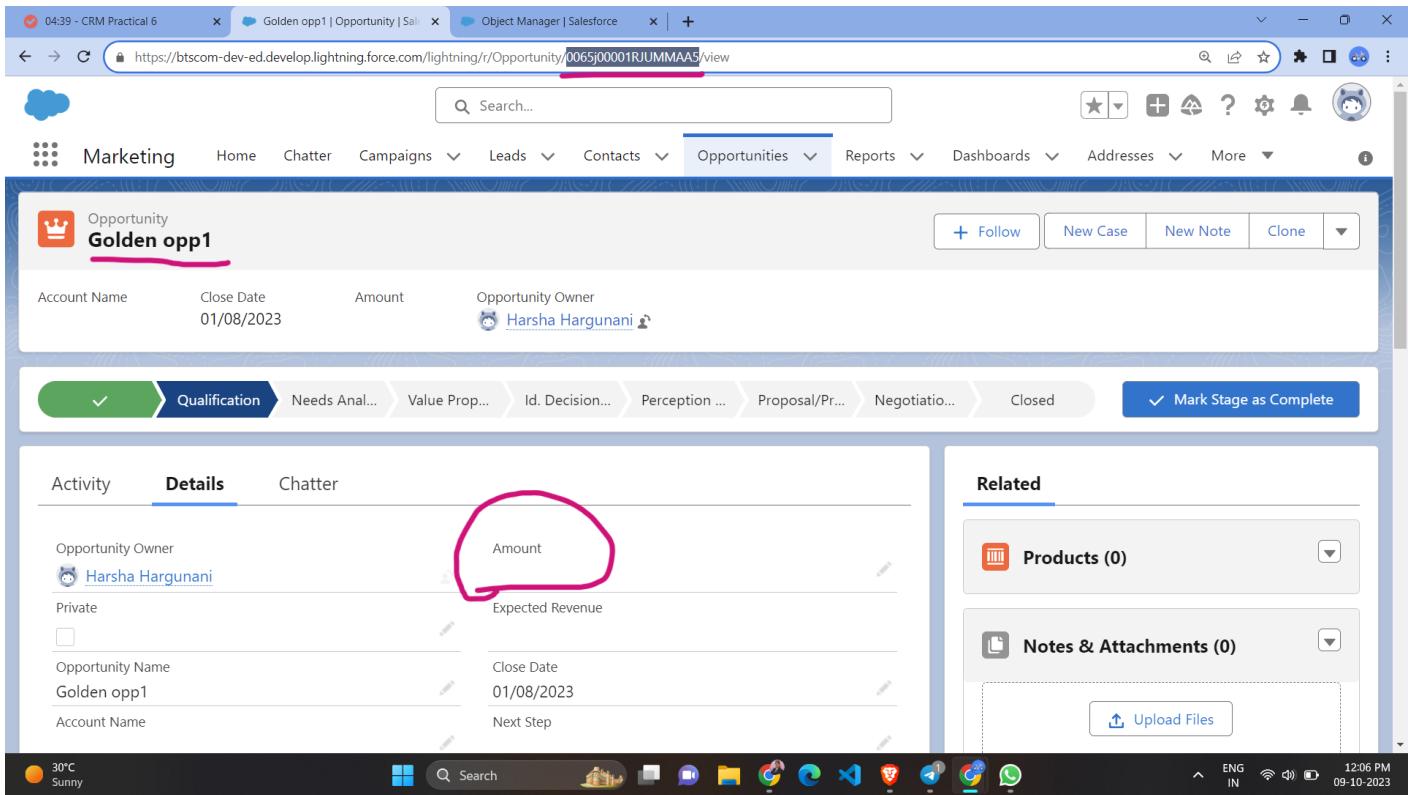
First Contact object with Last Name gets added successfully. While the second Contact object without a specified Last Name does not get added and displays a "Failed" message in Logs.

UPDATE

- While Updating Any Record By Data import Wizard, Data Loader, Workbench Or By Apex Programming Record ID of that Record is Necessary.

Q11. Write an Apex query to update the 'Amount' field of the Opportunity object in Salesforce?

Ans. Copy the Opportunity ID from any of the records, where the Amount field is empty.



The screenshot shows the Salesforce Opportunities page for an opportunity named 'Golden opp1'. The 'Amount' field is highlighted with a pink circle. The page includes sections for Activity, Details, Chatter, and Related items. The 'Details' section shows the Opportunity Owner as 'Harsha Hargunani'. The 'Related' section shows 'Products (0)' and 'Notes & Attachments (0)'. The bottom of the screen shows the Windows taskbar with various icons and system status.

Code:

```
Opportunity objOpp = New Opportunity();
objOpp.ID = '0065j00001RJUMMA5';
objOpp.Amount = 101.47;
update objOpp;
```

Enter Apex Code

```

1 Opportunity objOpp = New Opportunity();
2 objOpp.ID = '0065j00001RJUMMAA5';
3 objOpp.Amount = 101.47;
4 update objOpp;

```

Open Log

Output -

The screenshot shows a Salesforce Opportunity record for 'Golden opp1'. The key details are:

- Opportunity Name: Golden opp1
- Account Name: (empty)
- Close Date: 01/08/2023
- Amount: ₹101.47
- Opportunity Owner: Harsha Hargunani
- Status: Closed
- Stage: Qualification

The 'Details' tab is selected. In the 'Related' section, there are sections for Products (0) and Notes & Attachments (0). The system status bar at the bottom shows it's 12:11 PM on 09-10-2023.

Q12. Write an Apex query to update the Amount Field of multiple opportunity objects. One with Opportunity ID and another without it.

Ans. Before Execution:

All Opportunities

| | Opportunity Name | Account Name | Amount | Close Date | Stage | Opportunity Owner Alias |
|---|------------------|--------------|---------|------------|---------------|-------------------------|
| 1 | Golden opp1 | | ₹101.47 | 01/08/2023 | Qualification | HHarg |
| 2 | Golden opp2 | | | 01/08/2023 | Qualification | HHarg |
| 3 | Golden opp3 | | | 01/08/2023 | Qualification | HHarg |
| 4 | Golden opp4 | | | 01/08/2023 | Qualification | HHarg |
| 5 | Golden opp5 | | | 01/08/2023 | Qualification | HHarg |
| 6 | Golden opp6 | | | 01/08/2023 | Qualification | HHarg |
| 7 | Golden opp7 | | | 01/08/2023 | Qualification | HHarg |
| 8 | Golden opp8 | | | 01/08/2023 | Qualification | HHarg |

Code:

```
List<Opportunity> oppList = new List<Opportunity>();
```

```
Opportunity objOpp1 = New Opportunity();
```

```
objOpp1.ID = '0065j00001RJUMNAA5';
```

```
objOpp1.Amount = 201.53;
```

```
Opportunity objOpp2 = New Opportunity();
```

```
objOpp2.Amount = 87.90;
```

```
oppList.add(objOpp1);
```

```
oppList.add(objOpp2);
```

```
if(!oppList.isEmpty()) {
```

```
    Database.update(oppList, False);
```

```
}
```

Enter Apex Code

```

1 List<Opportunity> oppList = new List<Opportunity>();
2
3 Opportunity objOpp1 = New Opportunity();
4 objOpp1.ID = '0065j00001RJUMNAA5';
5 objOpp1.Amount = 201.53;
6
7 Opportunity objOpp2 = New Opportunity();
8 objOpp2.Amount = 87.90;
9
10 oppList.add(objOpp1);
11 oppList.add(objOpp2);
12
13 if(!oppList.isEmpty()){
14     Database.update(oppList, False);
15 }

```

Open Log Execute Execute Highlighted

Output:

| | Opportunity Name | Account Name | Amount | Close Date | Stage | Opportunity Owner Alias |
|---|------------------|--------------|---------|------------|---------------|-------------------------|
| 1 | Golden opp1 | | ₹101.47 | 01/08/2023 | Qualification | HHarg |
| 2 | Golden opp2 | | ₹201.53 | 01/08/2023 | Qualification | HHarg |
| 3 | Golden opp3 | | | 01/08/2023 | Qualification | HHarg |
| 4 | Golden opp4 | | | 01/08/2023 | Qualification | HHarg |
| 5 | Golden opp5 | | | 01/08/2023 | Qualification | HHarg |
| 6 | Golden opp6 | | | 01/08/2023 | Qualification | HHarg |
| 7 | Golden opp7 | | | 01/08/2023 | Qualification | HHarg |
| 8 | Golden opp8 | | | 01/08/2023 | Qualification | HHarg |

We observe that the Amount for "Golden opp3" does not get updated as the Opportunity ID was not specified. In order to avoid getting errors, we use the Database Update Method with boolean value False.

Database.update(oppList, False)

As a result, the update operation was performed partially, updating records for which we specified the ID in the code while leaving those for which we did not specify the ID untouched.

DELETE

Q13. Write an Apex code to delete multiple records.

Ans. Code:

```
List<Opportunity> oppList = New List<Opportunity>();  
Opportunity objOpp1 = New Opportunity();  
objOpp1.ID='0065j00001RJUMMAA5';  
Opportunity objOpp2 = New Opportunity();  
objOpp2.ID = '0065j00001RJUMNAA5';  
Opportunity objOpp3 = New Opportunity();  
objOpp3.ID = '0065j00001RJUMOAA5';  
Opportunity objOpp4 = New Opportunity();  
objOpp4.ID = '0065j00001RJUMPAA5';  
  
oppList.add(objOpp1);  
oppList.add(objOpp2);  
oppList.add(objOpp3);  
oppList.add(objOpp4);  
  
if(!oppList.isEmpty())  
    Database.delete(oppList,False);
```



The screenshot shows the 'Enter Apex Code' window from the Salesforce developer console. The code area contains the Apex code provided above, with line numbers 1 through 21 on the left. The code creates four Opportunity objects and adds them to a list, then deletes the list if it's not empty. At the bottom of the window, there are three buttons: 'Open Log' (with a checked checkbox), 'Execute', and 'Execute Highlighted'.

```
1 List<Opportunity> oppList = New List<Opportunity>();  
2  
3 Opportunity objOpp1 = New Opportunity();  
4 objOpp1.ID='0065j00001RJUMMAA5';  
5  
6 Opportunity objOpp2 = New Opportunity();  
7 objOpp2.ID = '0065j00001RJUMNAA5';  
8  
9 Opportunity objOpp3 = New Opportunity();  
10 objOpp3.ID = '0065j00001RJUMOAA5';  
11  
12 Opportunity objOpp4 = New Opportunity();  
13 objOpp4.ID = '0065j00001RJUMPAA5';  
14  
15 oppList.add(objOpp1);  
16 oppList.add(objOpp2);  
17 oppList.add(objOpp3);  
18 oppList.add(objOpp4);  
19  
20 if(!oppList.isEmpty())  
21     Database.delete(oppList,False);
```

Output: Search “Recycle Bin” in Apps.

The screenshot shows the Salesforce App Launcher interface. At the top, there is a search bar with the placeholder "Search..." and a magnifying glass icon. Below the search bar, the word "Marketing" is displayed next to a grid icon. To the right of "Marketing" are links for "Home", "Chatter", and "Campaigns". A large search result card is visible, showing a search bar with the text "rec" and a close button. The card has two main sections: "Apps" and "Items". Under "Apps", it says "No results". Under "Items", there are several items listed: "Duplicate Record Sets", "Forecasts", "Recommendations", "Recycle Bin" (which is highlighted in blue), "Scorecards", and "View All".

And we can find the 4 records deleted from Opportunity here. These records can also be restored or permanently deleted.

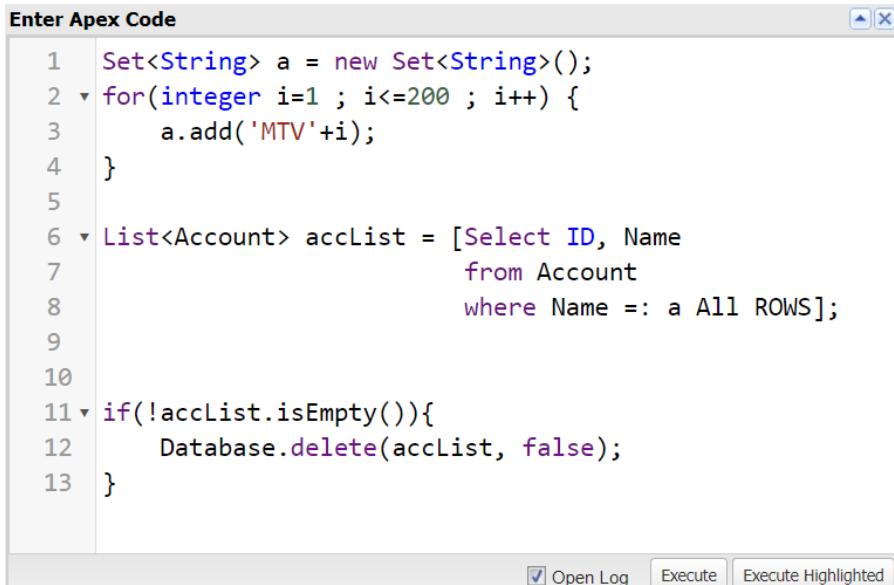
The screenshot shows the "My Recycle Bin" page in the Salesforce Lightning Experience. The browser tab is titled "My Recycle Bin | Recycle Bin | Sales [REDACTED]". The page header includes the "Marketing" navigation item and a "Search..." bar. On the right side of the header, there are buttons for "Restore", "Delete", and "Empty Org Recycle Bin". Below the header, a message states "6 items • Sorted by Record Name • Filtered by My recycle bin • Updated a few seconds ago". A table lists the deleted records:

| | Record Name | Type | Deleted By | Deleted Date |
|---|-------------|-------------|------------------|----------------------|
| 1 | GID-0000006 | Guest | Harsha Hargunani | 22/09/2023, 7:32 pm |
| 2 | GID-0000010 | Guest | Harsha Hargunani | 23/09/2023, 2:29 am |
| 3 | Golden opp1 | Opportunity | Harsha Hargunani | 09/10/2023, 12:29 pm |
| 4 | Golden opp2 | Opportunity | Harsha Hargunani | 09/10/2023, 12:29 pm |
| 5 | Golden opp3 | Opportunity | Harsha Hargunani | 09/10/2023, 12:29 pm |
| 6 | Golden opp4 | Opportunity | Harsha Hargunani | 09/10/2023, 12:29 pm |

At the bottom of the screen, there is a Windows taskbar showing various open applications and system status icons.

OR

If we want to delete records in bulk using SOQL query:



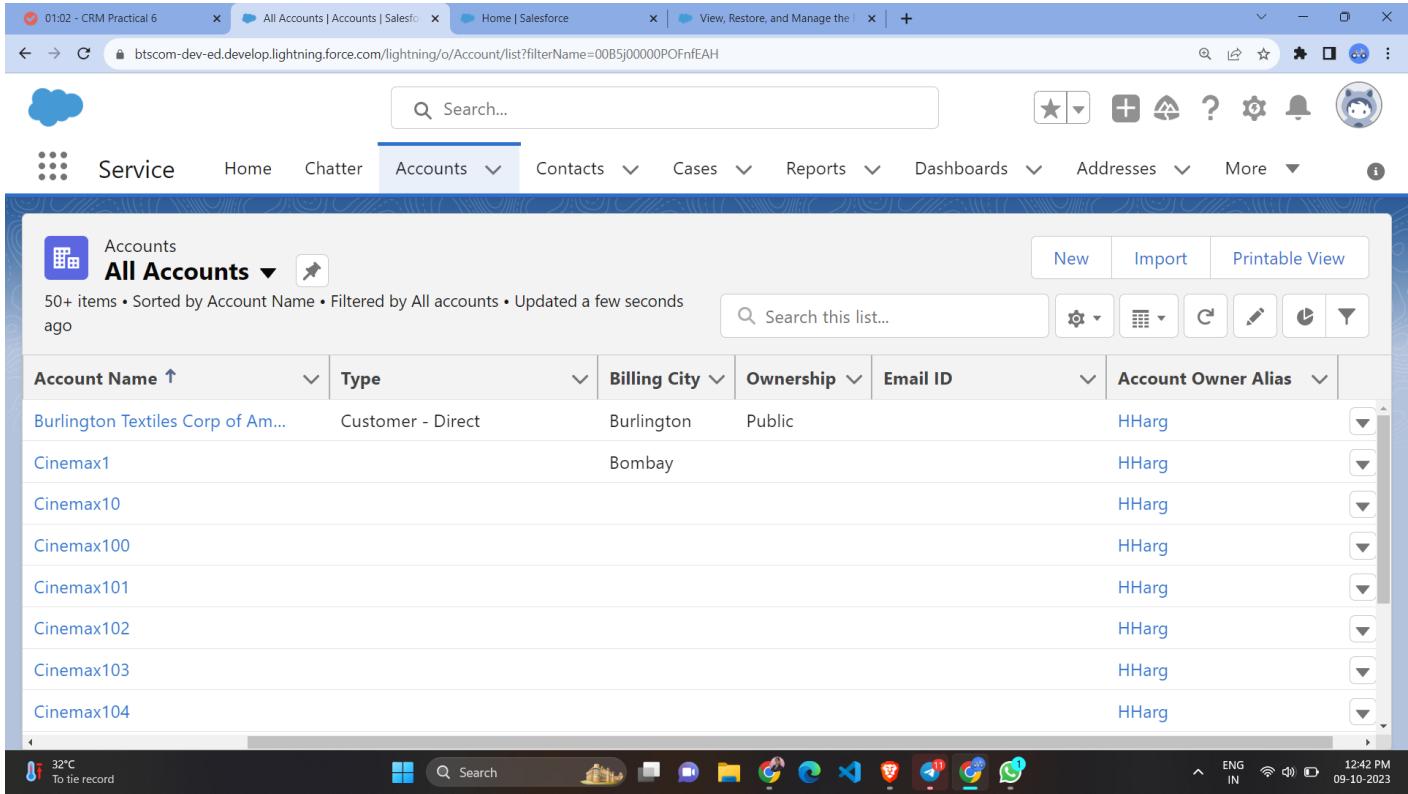
```
Enter Apex Code
1 Set<String> a = new Set<String>();
2 for(integer i=1 ; i<=200 ; i++) {
3     a.add('MTV'+i);
4 }
5
6 List<Account> accList = [Select ID, Name
7                             from Account
8                             where Name =: a All ROWS];
9
10
11 if(!accList.isEmpty()){
12     Database.delete(accList, false);
13 }
```

Open Log Execute Execute Highlighted

UPsert

Q14. Update Account Billing City From Bombay to Mumbai, and Insert new Account.

Ans. Before:



01:02 - CRM Practical 6 | All Accounts | Accounts | Salesforce | Home | Salesforce | View, Restore, and Manage the ... | +

CRM Practical 6

Service Home Chatter Accounts Contacts Cases Reports Dashboards Addresses More

Accounts All Accounts ▾

50+ items • Sorted by Account Name • Filtered by All accounts • Updated a few seconds ago

| Account Name ↑ | Type | Billing City | Ownership | Email ID | Account Owner Alias |
|-----------------------------------|-------------------|--------------|-----------|----------|---------------------|
| Burlington Textiles Corp of Am... | Customer - Direct | Burlington | Public | HHarg | |
| Cinemax1 | | Bombay | | HHarg | |
| Cinemax10 | | | | HHarg | |
| Cinemax100 | | | | HHarg | |
| Cinemax101 | | | | HHarg | |
| Cinemax102 | | | | HHarg | |
| Cinemax103 | | | | HHarg | |
| Cinemax104 | | | | HHarg | |

32°C To tie record

Search

12:42 PM 09-10-2023

Code:

```
Account[] acctsList = [SELECT Id, Name, BillingCity FROM Account WHERE BillingCity = 'Bombay'];
for (Account a: acctsList) {
    a.BillingCity = 'Mumbai';
}
Account newAcct = new Account(Name = 'Lee Mihee', BillingCity = 'Seoul');
acctsList.add(newAcct);
database.upsert(acctsList,false);
```

Enter Apex Code

```
1 Account[] acctList = [SELECT Id, Name, BillingCity FROM Account WHERE BillingCity = 'Bombay'];
2
3 for (Account a: acctList) {
4     a.BillingCity = 'Mumbai';
5 }
6
7 Account newAcct = new Account(Name = 'Lee Mihee', BillingCity = 'Seoul');
8 acctList.add(newAcct);
9
10 database.upsert(acctList, false);
```

Open Log

Output:

All Accounts | Accounts | Salesfo Home | Salesforce

05:00 - Time for a break! All Accounts | Accounts | Salesfo Home | Salesforce

btscom-dev-ed.develop.lightning.force.com/lightning/o/Account/list?filterName=00B5j00000POFnfEAH

Cloud Search... Service Home Chatter Accounts Contacts Cases Reports Dashboards Addresses More

Accounts All Accounts

50+ items • Sorted by Billing City • Filtered by All accounts • Updated a few seconds ago

| | <input type="checkbox"/> Account ... | Account Name | Type | Billing... | Ownership | Email ID | A |
|---|--------------------------------------|---------------------------------|--------------------|-------------|-----------|----------|---|
| 1 | <input type="checkbox"/> CD736025 | University of Arizona | Customer - Direct | Tucson | Other | | |
| 2 | <input type="checkbox"/> CD35512... | United Oil & Gas, Singapore | Customer - Direct | Singapore | Public | | |
| 3 | <input type="checkbox"/> | Lee Mihee | | Seoul | | | |
| 4 | <input type="checkbox"/> | sForce | | San Fran... | | | |
| 5 | <input type="checkbox"/> CC947211 | Express Logistics and Transport | Customer - Channel | Portland | Public | | |
| 6 | <input type="checkbox"/> CC213425 | Pyramid Construction Inc. | Customer - Channel | Paris | Public | | |
| 7 | <input type="checkbox"/> CD355118 | United Oil & Gas Corp. | Customer - Direct | New York | Public | | |
| 8 | <input type="checkbox"/> | Cinemax1 | | Mumbai | | | |

35°C Haze ENG IN 02:56 PM 09-10-2023

Q15. Write Apex code Update Account Type from Prospect to Customer – Direct and Insert new Accounts Names -Blue, red, green, orange.

Ans. Code:

```
List<Account> accList=[Select Id,Name,Type from Account Where type='Prospect'];
if(!accList.isEmpty()){
    For(Account a:accList){
        system.debug('Account Name= '+a.name);
        system.debug('type= '+a.type);
        a.type='Customer – Direct';
        system.debug('updated type= '+a.type);
    }
}
List<Account> newAccList=new List<Account>();
Account objAccount1=new Account();
objAccount1.Name='Blue';

Account objAccount2=new Account();
objAccount2.Name='Red';

Account objAccount3=new Account();
objAccount3.Name='Green';

Account objAccount4=new Account();
objAccount4.Name='Orange';

newAccList.add(objAccount1);
newAccList.add(objAccount2);
newAccList.add(objAccount3);
newAccList.add(objAccount4);

accList.addAll(newAccList);
database.upsert(accList,False);
```

Q16. Write Apex Code to Update Applicant Police Verification from True to False and insert new Applicant Names from list given by client.

Ans. Code:

```
List<Applicant__c> appList=[Select ID,First_Name__c,Police_Verification__c from Applicant__c where
Police_Verification__c=True];
if(!appList.isEmpty()){

}
```

```

for(Applicant__c a:appList){
    system.debug('First Name= '+a.First_Name__c);
    system.debug('Previous Police Verification= '+a.Police_Verification__c);
    a.Police_Verification__c=False;
    system.debug('Updated Police Verification= '+a.Police_Verification__c);

}
}

string List_Given_By_Client='Tom,Tek,Harry,John,Jony,Janadan,Amar,Akbar,Anthony';
List<String> nameList=List_Given_By_Client.Split(',');

List<Applicant__c> newAppList=new List<Applicant__c>{};

For(String n:nameList){
    Applicant__c objApplicant=new Applicant__c();
    objApplicant.First_Name__c=n;
    newAppList.add(objApplicant);
}
appList.AddAll(newAppList);

if(!appList.isEmpty())
database.upsert(appList);

```

UNDELETE

- After you have deleted records, the records are placed in the Recycle Bin for 15 days, after which they are permanently deleted. While the records are still in the Recycle Bin, you can restore them using the undelete operation.
- If you accidentally deleted some records that you want to keep, restore them from the Recycle Bin.
- All Rows- The ALL ROWS keyword queries all rows for both top level and aggregate relationships, including deleted records and archived activities.

Q17. Use undelete operation to restore deleted both Parent and Child records associated in a Master-Detail Relationship.

Ans. Step1: Delete a record on Master side such that the child records get cascadingly gets deleted.

Parent Object - Student Details

10:51 - CRM Practical 6 | All | Students Details | Salesforce | All | Subjects | Salesforce | Home | Salesforce

btscm-dev-ed.develop.lightning.force.com/lightning/o/Student_Details_c/list?filterName=00B5j00000Oj5LIEAZ

Passport Cloud Home Dashboards Customers Chatter Reports

Students Details All

4 items • Sorted by StudentID • Filtered by All students details • Updated 2 minutes ago

| | StudentID ↑ | First Name | Last Name |
|---|-------------|------------|-----------|
| 1 | 102 | Seokjin | Kim |
| 2 | 106 | Jungkook | Jeon |
| 3 | 107 | Namjoon | Kim |
| 4 | 108 | Yuna | Shin |

Search... Search this list...

Harsha Hargunani
btscm-dev-ed.develop.my.salesforce.com
[Settings](#) [Log Out](#)

USERNAMES

shreemodi@regionalmanager.com
btscm-dev-ed.develop.my.salesforce.com

krishnamodi@agent.com

DISPLAY DENSITY

Comfy

Compact

OPTIONS

[Switch to Salesforce Classic](#)

Add Username

35°C Haze

Search

03:11 PM 09-10-2023

Child Object - Subjects

10:21 - CRM Practical 6 | All | Students Details | Salesforce | All | Subjects | Salesforce | Home | Salesforce

btscm-dev-ed.develop.lightning.force.com/lightning/o/Subject_c/list?filterName=00B5j00000OkdtEAR

Passport Cloud Home Dashboards Customers Chatter Reports

Subjects All

2 items • Sorted by Subject Name • Filtered by All subjects • Updated a few seconds ago

| | Subject Name ↑ | Student Details |
|---|-------------------|-----------------|
| 1 | Cloud Computing | 102 |
| 2 | Operating Systems | 102 |

Search... Search this list...

Harsha Hargunani
btscm-dev-ed.develop.my.salesforce.com
[Settings](#) [Log Out](#)

USERNAMES

shreemodi@regionalmanager.com
btscm-dev-ed.develop.my.salesforce.com

krishnamodi@agent.com

DISPLAY DENSITY

Comfy

Compact

OPTIONS

[Switch to Salesforce Classic](#)

Add Username

35°C Haze

Search

03:12 PM 09-10-2023

We delete StudentID - 102, Student Name - Kim Seokjin

Students Details

All

| | StudentID ↑ | First Name | Last Name | Gender |
|---|-------------|------------|-----------|--------|
| 1 | 102 | Seokjin | Kim | Male |
| 2 | 106 | Jungkook | Jeon | |
| 3 | 107 | Namjoon | Kim | Others |
| 4 | 108 | Yuna | Shin | Female |

4 items • Sorted by StudentID • Filtered by All students details • Updated 3 minutes ago

Search this list...

New Import Change Owner Printable View

Edit

Delete

Change Owner

javascript:void(0);

35°C Haze

03:13 PM 09-10-2023

Student Details "102" was deleted. [Undo](#)

Change Owner Printable View

3 items • Sorted by StudentID • Filtered by All students details • Updated a few seconds ago

| | StudentID ↑ | First Name | Last Name | Gender |
|---|-------------|------------|-----------|--------|
| 1 | 106 | Jungkook | Jeon | |
| 2 | 107 | Namjoon | Kim | Others |
| 3 | 108 | Yuna | Shin | Female |

03:13 PM 09-10-2023

Both Cloud Computing and OS Subjects get cascade delete.

The screenshot shows the Passport Cloud application interface. At the top, there are four tabs: "07:36 - CRM Practical 6", "All | Students Details | Salesforce", "All | Subjects | Salesforce", and "Home | Salesforce". Below the tabs is a search bar with placeholder text "Search...". The main header includes the "Passport Cloud" logo, navigation links for Home, Dashboards, Customers, Chatter, Reports, Addresses, Subjects (which is currently selected), More, and a gear icon for settings. A blue sidebar on the left contains icons for Subjects, All, and a plus sign. The main content area displays a table with columns: "Subject Name ↑", "Student Details", and "Record ID ↑". A message at the bottom states "No items to display." The bottom of the screen shows a taskbar with various application icons and system status indicators like weather (35°C Haze), date (09-10-2023), and time (03:14 PM).

This Master record can be found in the Recycle Bin.

The screenshot shows the Service application interface. At the top, there are four tabs: "05:54 - CRM Practical 6", "All | Students Details | Salesforce", "Recently Viewed | Cars | Salesfor", and "My Recycle Bin | Recycle Bin | Sa". Below the tabs is a search bar with placeholder text "Search...". The main header includes the "Service" logo, navigation links for Home, Chatter, Accounts, Contacts, Cases, Reports, and a "My Recycle Bin" link which is currently selected. The main content area displays a table with columns: "Record Name ↑", "Type", "Deleted By", and "Deleted Date". The table lists 7 items, all deleted by "Harsha Hargunani" on various dates between 09/10/2023 and 23/09/2023. The bottom of the screen shows a taskbar with various application icons and system status indicators like weather (35°C Haze), date (09-10-2023), and time (03:16 PM).

Code:

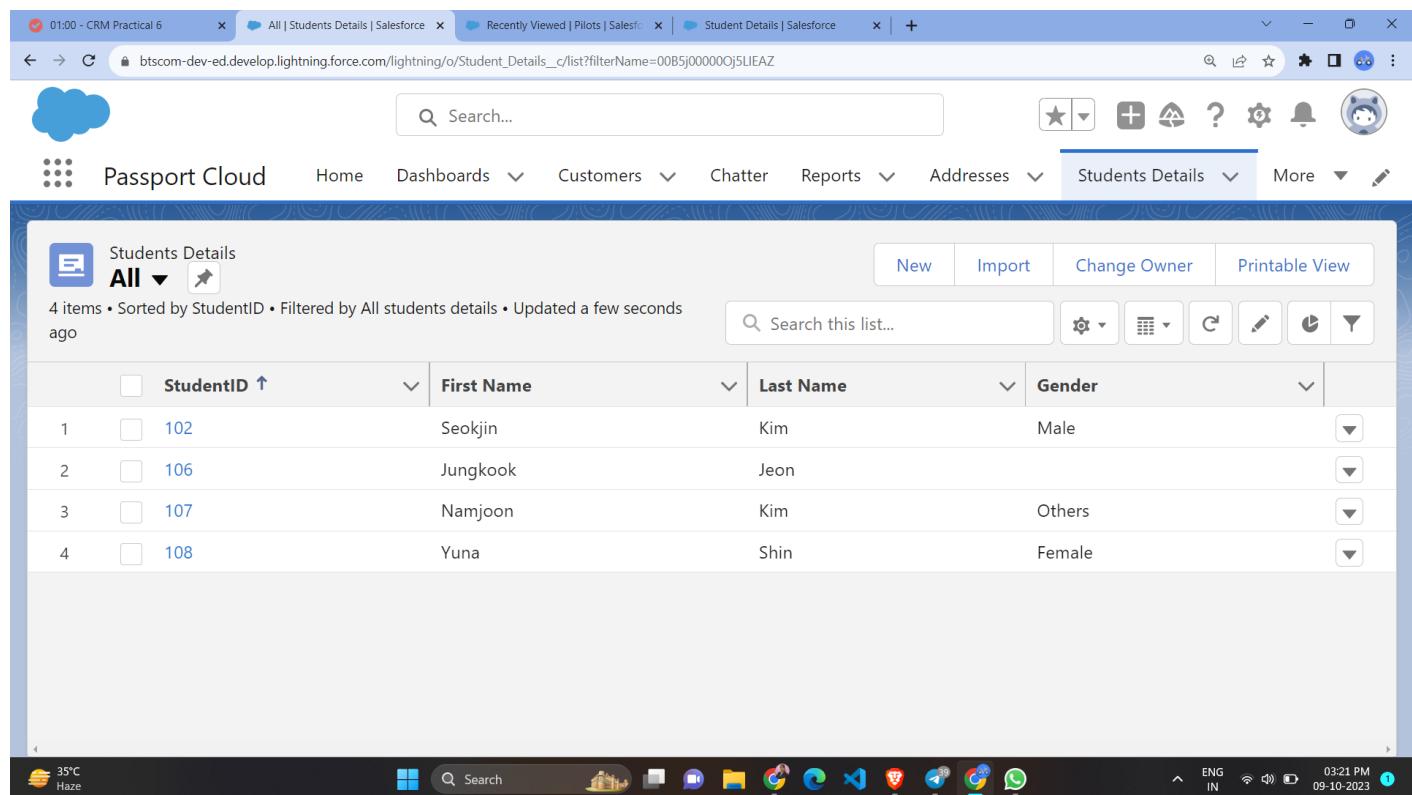
```
List<Student_Details__c> studList = [select Name, First_Name__c  
    from Student_Details__c  
    where First_Name__c = 'Seokjin'  
    ALL ROWS];  
  
if(!studList.isEmpty()){  
    Database.undelete(studList, False);  
}
```

Enter Apex Code

```
1 List<Student_Details__c> studList = [select Name, First_Name__c  
2         from Student_Details__c  
3         where First_Name__c = 'Seokjin'  
4         ALL ROWS];  
5 if(!studList.isEmpty()){  
6     Database.undelete(studList, False);  
7 }
```

Open Log Execute Execute Highlighted

Output: Student record and associated child (subjects) records were restored.



01:00 - CRM Practical 6 All | Students Details | Salesforce Recently Viewed | Pilots | Salesfc Student Details | Salesforce

Search... Open Log Execute Execute Highlighted

Passport Cloud Home Dashboards Customers Chatter Reports Addresses Students Details More

| | StudentID ↑ | First Name | Last Name | Gender | |
|---|-------------|------------|-----------|--------|---|
| 1 | 102 | Seokjin | Kim | Male | ▼ |
| 2 | 106 | Jungkook | Jeon | Others | ▼ |
| 3 | 107 | Namjoon | Kim | Others | ▼ |
| 4 | 108 | Yuna | Shin | Female | ▼ |

35°C Haze Search 03:21 PM 09-10-2023

00:24 - CRM Practical 6

All | Subjects | Salesforce

Recently Viewed | Pilots | Salesfo

btscom-dev-ed.develop.lightning.force.com/lightning/o/Subject_c/list?filterName=00B5j00000OkdHEAR

Passport Cloud

Subjects

All

New Import Printable View

Search this list...

Subject Name ↑ Student Details Record ID ↑

| | | | |
|---|--|-----|--------------------|
| 1 | <input type="checkbox"/> Cloud Computing | 102 | a0A5j00000EkuEgEAJ |
| 2 | <input type="checkbox"/> Operating Systems | 102 | a0A5j00000EISbIEAV |

35°C Haze

Search ENG IN 03:22 PM 09-10-2023

Q18. Write Apex program to restore multiple deleted records using Database undelete method.

Ans. Delete 4 records and check the Recycle Bin.

| | | | | | |
|---|--------------------------------------|-------------|------------------|----------------------|----------------------------------|
| 3 | <input type="checkbox"/> Golden opp1 | Opportunity | Harsha Hargunani | 09/10/2023, 12:29 pm | <input type="button" value="▼"/> |
| 4 | <input type="checkbox"/> Golden opp2 | Opportunity | Harsha Hargunani | 09/10/2023, 12:29 pm | <input type="button" value="▼"/> |
| 5 | <input type="checkbox"/> Golden opp3 | Opportunity | Harsha Hargunani | 09/10/2023, 12:29 pm | <input type="button" value="▼"/> |
| 6 | <input type="checkbox"/> Golden opp4 | Opportunity | Harsha Hargunani | 09/10/2023, 12:29 pm | <input type="button" value="▼"/> |

Code:

```
Set<String> s = new Set<String>();
for(integer i=1 ; i<=4 ; i++) {
    s.add('Golden opp'+i);
}
```

```
List<Opportunity> oppList = [Select ID, Name
                            from Opportunity
                            where Name =: s All ROWS];
```

```
if(!oppList.isEmpty()){
    Database.undelete(oppList, false);
}
```

Enter Apex Code

```

1 Set<String> s = new Set<String>();
2 for(integer i=1 ; i<=4 ; i++) {
3     s.add('Golden opp'+i);
4 }
5
6 List<Opportunity> oppList = [Select ID, Name
7                                 from Opportunity
8                                 where Name =: s All ROWS];
9
10
11 if(!oppList.isEmpty()){
12     Database.undelete(oppList, false);
13 }

```

Open Log Execute Execute Highlighted

Output: All 4 records were restored and can no longer be seen in Recycle Bin.

| | Opportunity Name | Account Name | Amount | Close Date | Stage | Opportunity Owner Alias |
|---|------------------|--------------|---------|------------|---------------|-------------------------|
| 1 | Golden opp1 | | ₹101.47 | 01/08/2023 | Qualification | HHarg |
| 2 | Golden opp2 | | ₹201.53 | 01/08/2023 | Qualification | HHarg |
| 3 | Golden opp3 | | | 01/08/2023 | Qualification | HHarg |
| 4 | Golden opp4 | | | 01/08/2023 | Qualification | HHarg |
| 5 | Golden opp5 | | | 01/08/2023 | Qualification | HHarg |
| 6 | Golden opp6 | | | 01/08/2023 | Qualification | HHarg |
| 7 | Golden opp7 | | | 01/08/2023 | Qualification | HHarg |
| 8 | Golden opp8 | | | 01/08/2023 | Qualification | HHarg |

SOQL- Salesforce Object Query Language

- When to use SOQL- if we are required to Fetch/display Some Records from objects in Anonymous Window or in a Class or anywhere then we use SOQL.

- Syntax - List<Object> ListName = [Select ColumnName1, ColumnName2,, ColumnNameN from ObjectName];
- In SQL, the * wildcard can be used to fetch all fields in a query. However, Salesforce operates in a multitenant environment, and as a result, SOQL does not support the use of the "*" wildcard for selecting all fields.

Q19. Write a SOQL query to display all student names.

Ans. Code:

```
List<Student_Details__c> sList = [Select First_Name__c,
                                     Last_Name__c, Gender__c
                                     From Student_Details__c];
if(!sList.isEmpty()){
    For(Student_Details__c st:sList){
        system.debug('Name = ' + st.First_Name__c +
                     + st.Last_Name__c + ', Gender = '
                     + st.Gender__c);
    }
}
```

Enter Apex Code

```
1  List<Student_Details__c> sList = [Select First_Name__c,
2                                     Last_Name__c, Gender__c
3                                     From Student_Details__c];
4  if(!sList.isEmpty()){
5      For(Student_Details__c st:sList){
6          system.debug('Name = ' + st.First_Name__c +
7                      + st.Last_Name__c + ', Gender = '
8                      + st.Gender__c);
9      }
10 }
```

Open Log

Output:

File ▼ Edit ▼ Debug ▼ Test ▼ Workspace ▼ Help ▼ < >

Log executeAnonymous @10/9/2023, 4:02:19 PM

Execution Log

| Timestamp | Event | Details |
|--------------|------------|---|
| 16:02:19:022 | USER_DEBUG | [6] DEBUG Name = Jungkook Jeon, Gender = null |
| 16:02:19:022 | USER_DEBUG | [6] DEBUG Name = Namjoon Kim, Gender = Others |
| 16:02:19:022 | USER_DEBUG | [6] DEBUG Name = Seokjin Kim, Gender = Male |
| 16:02:19:023 | USER_DEBUG | [6] DEBUG Name = Yuna Shin, Gender = Female |

Q20. Write Apex query to fetch 10 Account records using Limit keyword.

Ans. Code:

```
List<Account> accList = [Select Name From Account LIMIT 10];
system.debug('Size = ' + accList.size());
if(!accList.isEmpty()){
    For(Account a:accList){
        system.debug('Name = ' + a.Name);
    }
}
```

Enter Apex Code

```
1 List<Account> accList = [Select Name
2                     From Account
3                     LIMIT 10];
4
5 system.debug('Size= '+accList.size());
6
7 if(!accList.isEmpty()){
8     For(Account a:accList){
9         system.debug('Name = ' + a.Name);
10    }
11 }
```

Open Log Execute Execute Highlighted

Output:

Log executeAnonymous @10/9/2023, 4:11:40 PM

| Execution Log | | |
|---------------|------------|----------------------------|
| Timestamp | Event | Details |
| 16:11:40:016 | USER_DEBUG | [5] DEBUG Size= 10 |
| 16:11:40:016 | USER_DEBUG | [9] DEBUG Name = Cinemax16 |
| 16:11:40:016 | USER_DEBUG | [9] DEBUG Name = Cinemax17 |
| 16:11:40:016 | USER_DEBUG | [9] DEBUG Name = Cinemax18 |
| 16:11:40:016 | USER_DEBUG | [9] DEBUG Name = Cinemax19 |
| 16:11:40:016 | USER_DEBUG | [9] DEBUG Name = Cinemax20 |
| 16:11:40:017 | USER_DEBUG | [9] DEBUG Name = Cinemax21 |
| 16:11:40:017 | USER_DEBUG | [9] DEBUG Name = Cinemax22 |
| 16:11:40:017 | USER_DEBUG | [9] DEBUG Name = Cinemax23 |
| 16:11:40:017 | USER_DEBUG | [9] DEBUG Name = Cinemax24 |
| 16:11:40:017 | USER_DEBUG | [9] DEBUG Name = Cinemax25 |

Child To Parent Traverse

Q21. Write Apex code to display Contact Name With Its Account Name.

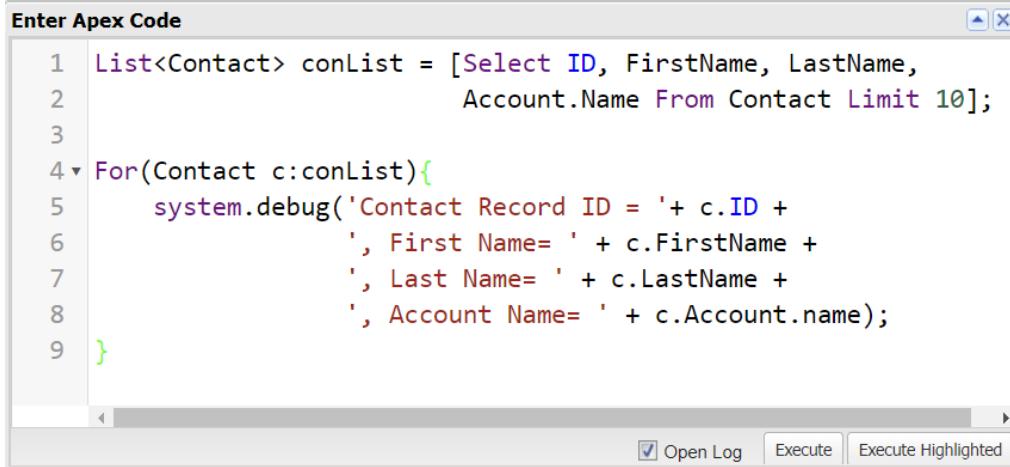
Ans. Code:

```
List<Contact> conList = [Select ID, FirstName, LastName, Account.Name From Contact Limit 10];
```

```
For(Contact c:conList){
```

```
    system.debug('Contact Record ID = '+ c.ID +  
        ', First Name= ' + c.FirstName +  
        ', Last Name= ' + c.LastName +  
        ', Account Name= ' + c.Account.name);
```

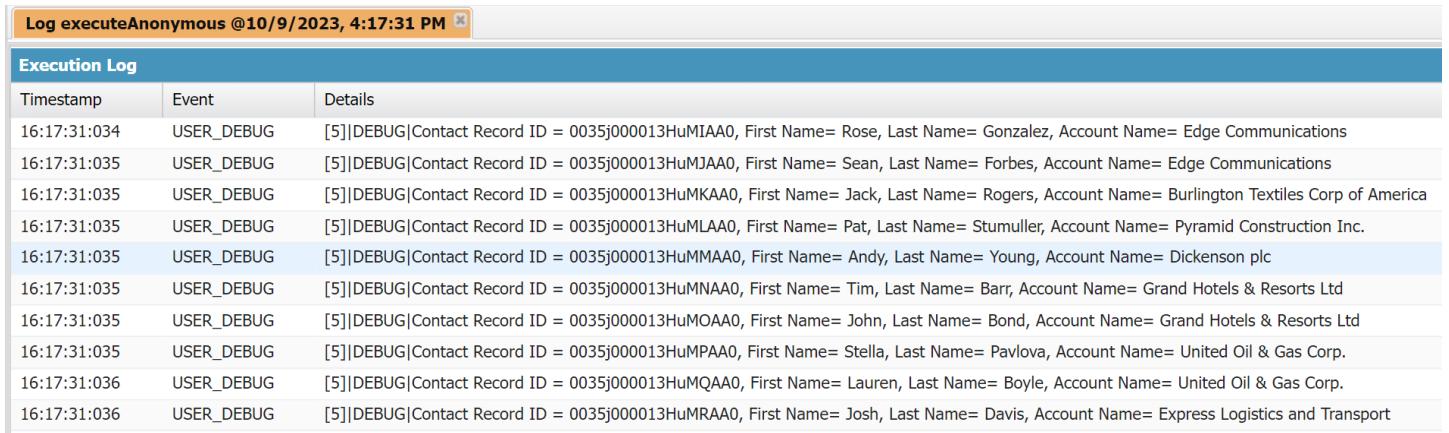
```
}
```



```
1 List<Contact> conList = [Select ID, FirstName, LastName,  
2                             Account.Name From Contact Limit 10];  
3  
4 For(Contact c:conList){  
5     system.debug('Contact Record ID = '+ c.ID +  
6         ', First Name= ' + c.FirstName +  
7         ', Last Name= ' + c.LastName +  
8         ', Account Name= ' + c.Account.name);  
9 }
```

The screenshot shows the 'Enter Apex Code' window in the Salesforce IDE. The code editor contains the Apex code provided above. At the bottom of the window, there are three buttons: 'Open Log' (with a checked checkbox), 'Execute', and 'Execute Highlighted'.

Output:



| Execution Log | | |
|---------------|------------|--|
| Timestamp | Event | Details |
| 16:17:31:034 | USER_DEBUG | [5] DEBUG Contact Record ID = 0035j000013HuMIAA0, First Name= Rose, Last Name= Gonzalez, Account Name= Edge Communications |
| 16:17:31:035 | USER_DEBUG | [5] DEBUG Contact Record ID = 0035j000013HuMJAA0, First Name= Sean, Last Name= Forbes, Account Name= Edge Communications |
| 16:17:31:035 | USER_DEBUG | [5] DEBUG Contact Record ID = 0035j000013HuMKAA0, First Name= Jack, Last Name= Rogers, Account Name= Burlington Textiles Corp of America |
| 16:17:31:035 | USER_DEBUG | [5] DEBUG Contact Record ID = 0035j000013HuMLAA0, First Name= Pat, Last Name= Stummller, Account Name= Pyramid Construction Inc. |
| 16:17:31:035 | USER_DEBUG | [5] DEBUG Contact Record ID = 0035j000013HuMMAA0, First Name= Andy, Last Name= Young, Account Name= Dickenson plc |
| 16:17:31:035 | USER_DEBUG | [5] DEBUG Contact Record ID = 0035j000013HuMNAA0, First Name= Tim, Last Name= Barr, Account Name= Grand Hotels & Resorts Ltd |
| 16:17:31:035 | USER_DEBUG | [5] DEBUG Contact Record ID = 0035j000013HuMOAA0, First Name= John, Last Name= Bond, Account Name= Grand Hotels & Resorts Ltd |
| 16:17:31:035 | USER_DEBUG | [5] DEBUG Contact Record ID = 0035j000013HuMPAA0, First Name= Stella, Last Name= Pavlova, Account Name= United Oil & Gas Corp. |
| 16:17:31:036 | USER_DEBUG | [5] DEBUG Contact Record ID = 0035j000013HuMQAA0, First Name= Lauren, Last Name= Boyle, Account Name= United Oil & Gas Corp. |
| 16:17:31:036 | USER_DEBUG | [5] DEBUG Contact Record ID = 0035j000013HuMRRA0, First Name= Josh, Last Name= Davis, Account Name= Express Logistics and Transport |

Q22. Write Apex code to Display 20 Opportunities Name With its Account Name & Account Type.

Ans. Code:

```
List<Opportunity> oppList = [Select Name, Account.name, Account.type From Opportunity Limit 20];
```

```

For(Opportunity a:oppList){
    system.debug('Opportunity Name = ' + a.Name + ', Account Name = '
        + a.Account.Name + ', Account Type = ' + a.Account.Type);
}

```

Enter Apex Code

```

1 List<Opportunity> oppList = [Select Name, Account.name, Account.type
                                From Opportunity Limit 20];
2
3
4 For(Opportunity a:oppList){
5     system.debug('Opportunity Name = ' + a.Name + ', Account Name = '
6         + a.Account.Name + ', Account Type = ' + a.Account.Type);
7 }

```

Open Log

Output:

| Execution Log | | |
|---------------|------------|--|
| Timestamp | Event | Details |
| 16:42:47:024 | USER_DEBUG | [5] DEBUG Opportunity Name = Edge Emergency Generator, Account Name = Edge Communications, Account Type = Customer - Direct |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = Edge Installation, Account Name = Edge Communications, Account Type = Customer - Direct |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = Edge SLA, Account Name = Edge Communications, Account Type = Customer - Direct |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = Edge Emergency Generator, Account Name = Edge Communications, Account Type = Customer - Direct |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = Burlington Textiles Weaving Plant Generator, Account Name = Burlington Textiles Corp of America, Account Type = Customer - Direct |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = Pyramid Emergency Generators, Account Name = Pyramid Construction Inc., Account Type = Customer - Channel |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = Dickenson Mobile Generators, Account Name = Dickenson plc, Account Type = Customer - Channel |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = Grand Hotels Kitchen Generator, Account Name = Grand Hotels & Resorts Ltd, Account Type = Customer - Direct |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = Grand Hotels Guest Portable Generators, Account Name = Grand Hotels & Resorts Ltd, Account Type = Customer - Direct |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = Grand Hotels Generator Installations, Account Name = Grand Hotels & Resorts Ltd, Account Type = Customer - Direct |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = Grand Hotels SLA, Account Name = Grand Hotels & Resorts Ltd, Account Type = Customer - Direct |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = Grand Hotels Emergency Generators, Account Name = Grand Hotels & Resorts Ltd, Account Type = Customer - Direct |
| 16:42:47:025 | USER_DEBUG | [5] DEBUG Opportunity Name = United Oil SLA, Account Name = United Oil & Gas Corp., Account Type = Customer - Direct |
| 16:42:47:026 | USER_DEBUG | [5] DEBUG Opportunity Name = United Oil Office Portable Generators, Account Name = United Oil & Gas Corp., Account Type = Customer - Direct |
| 16:42:47:026 | USER_DEBUG | [5] DEBUG Opportunity Name = United Oil Refinery Generators, Account Name = United Oil & Gas Corp., Account Type = Customer - Direct |
| 16:42:47:026 | USER_DEBUG | [5] DEBUG Opportunity Name = United Oil Installations, Account Name = United Oil & Gas Corp., Account Type = Customer - Direct |
| 16:42:47:026 | USER_DEBUG | [5] DEBUG Opportunity Name = United Oil Installations, Account Name = United Oil & Gas Corp., Account Type = Customer - Direct |
| 16:42:47:026 | USER_DEBUG | [5] DEBUG Opportunity Name = United Oil Refinery Generators, Account Name = United Oil & Gas Corp., Account Type = Customer - Direct |
| 16:42:47:026 | USER_DEBUG | [5] DEBUG Opportunity Name = United Oil Installations, Account Name = United Oil & Gas Corp., Account Type = Customer - Direct |
| 16:42:47:026 | USER_DEBUG | [5] DEBUG Opportunity Name = United Oil Emergency Generators, Account Name = United Oil & Gas Corp., Account Type = Customer - Direct |

Q23. Write Apex code to display Custom Child object records along with its Parent Object data.

Ans. Child - Subjects (Subject Name, Student Details, Created By)

Parent - Student Details (StudentID, Student Full Name, Gender, DOB)

Fields & Relationships
4 Items, Sorted by Field Label

| FIELD LABEL | FIELD NAME | DATA TYPE |
|------------------|--------------------|--------------------------------|
| Created By | CreatedById | Lookup(User) |
| Last Modified By | LastModifiedById | Lookup(User) |
| Student Details | Student_Details__c | Master-Detail(Student Details) |
| Subject Name | Name | Text(80) |

Fields & Relationships
20 Items, Sorted by Field Label

| | | |
|-------------------|----------------------|------------------------------------|
| Last Name | Last_Name__c | Text(35) |
| My Number | My_Number__c | Formula (Number) |
| Owner | OwnerId | Lookup(User,Group) |
| PAN Number | PAN_Number__c | Text(10) (Unique Case Insensitive) |
| Phone No | Phone_No__c | Phone |
| Student Full Name | Student_Full_Name__c | Formula (Text) |
| StudentID | Name | Auto Number |

Code:

```
List<Subject__c> studList = [Select Name, Student_Details__r.Name,
                           Student_Details__r.Student_Full_Name__c
                           From Subject__c];
```

```

For(Subject__c s:studList){
    system.debug('Subject Name = ' + s.Name + ', Student ID = '
        + s.Student_Details__r.Name + ', Student Name = '
        + s.Student_Details__r.Student_Full_Name__c);
}

```

Enter Apex Code

```

1 List<Subject__c> studList = [Select Name, Student_Details__r.Name,
2                               Student_Details__r.Student_Full_Name__c
3                               From Subject__c];
4
5 For(Subject__c s:studList){
6     system.debug('Subject Name = ' + s.Name + ', Student ID = '
7         + s.Student_Details__r.Name + ', Student Name = '
8         + s.Student_Details__r.Student_Full_Name__c);
9 }

```

Open Log

Here, within the Child Object, we see the reference "Student_Details__c" API Name that connects it with Student Details. We change the "__c" to "__r" and use it to access Parent Object Fields.

Output:

Log executeAnonymous @10/9/2023, 8:01:04 PM

Execution Log

| Timestamp | Event | Details |
|--------------|------------|---|
| 20:01:04:015 | USER_DEBUG | [6] DEBUG Subject Name = MERN Stack Development, Student ID = 107, Student Name = Kim Namjoon |
| 20:01:04:015 | USER_DEBUG | [6] DEBUG Subject Name = Deep Learning, Student ID = 106, Student Name = Jeon Jungkook |
| 20:01:04:015 | USER_DEBUG | [6] DEBUG Subject Name = Artificial Intelligence, Student ID = 108, Student Name = Shin Yuna |
| 20:01:04:015 | USER_DEBUG | [6] DEBUG Subject Name = Cloud Computing, Student ID = 102, Student Name = Kim Seokjin |
| 20:01:04:015 | USER_DEBUG | [6] DEBUG Subject Name = Operating Systems, Student ID = 102, Student Name = Kim Seokjin |

Writing this query under “Query Editor” and hitting “Execute” button:

The screenshot shows the Salesforce Query Editor interface. At the top, there are buttons for "Query Grid", "Save Rows", "Insert Row", "Delete Row", and "Refresh Grid". Below these are tabs for "Logs", "Tests", "Checkpoints", "Query Editor" (which is highlighted in orange), "View State", "Progress", and "Problems". The main area contains the following SOQL query:

```
Select Name, Student_Details__r.Name, Student_Details__r.Student_Full_Name__c From Subject__c
```

Below the query, a message says "Any query errors will appear here...". At the bottom, there are buttons for "Execute" and "Use Tooling API".

The screenshot shows the Salesforce Query Editor interface after executing the query. The title bar says "Subject__c@8:09 PM". The query is displayed again:

```
Select Name, Student_Details__r.Name, Student_Details__r.Student_Full_Name__c From Subject__c
```

A "Query Results - Total Rows: 5" section shows the following data:

| Name | Student_Details__r.Name | Student_Details__r.Student_Full_Name__c |
|-------------------------|-------------------------|---|
| MERN Stack Development | 107 | Kim Namjoon |
| Deep Learning | 106 | Jeon Jungkook |
| Artificial Intelligence | 108 | Shin Yuna |
| Cloud Computing | 102 | Kim Seokjin |
| Operating Systems | 102 | Kim Seokjin |

Q24. Write Apex code with SOQL query to display Contact and Account Name having no email id exists in the contact and created in the range of 1 December 2020 to 5 January 2021.

Ans. Code:

```
List<Contact> conList=[Select Account.Name,Name,Email,CreatedDate from Contact where Email=null and CreatedDate> 2021-12-01T00:00:00Z and CreatedDate< 2022-01-10T00:00:00Z];
```

```
if(!conList.isEmpty()){
    for(Contact c:conList){
        system.debug('Account Name = ' + c.Account.Name + ' Name = ' + c.Name + ' Email ID = ' +
c.Email + ' Created Date = ' + c.CreatedDate);
    }
}
```

```
}
```

Enter Apex Code

```
1 ▼ List<Contact> conList = [Select Account.Name, Name, Email,
                                CreatedDate from Contact
                                where Email = null
                                and CreatedDate > 2021-12-01T00:00:00Z
                                and CreatedDate < 2022-01-10T00:00:00Z];
2
3
4
5
6
7 ▼ if(!conList.isEmpty()){
8   for(Contact c:conList){
9     system.debug('Account Name = '+ c.Account.Name + ', Name = '
10      + c.Name + ' Email ID = ' + c.Email +
11      ', Created Date = ' + c.CreatedDate);
12   }
13 }
```

Open Log Execute Execute Highlighted

Parent To Child Traverse

Q25. Write Apex program to display list of Accounts with Its Contacts.

Ans. Code:

```
List<Account> accList = [Select Name, (select FirstName, LastName From contacts) From Account];
For(Account a:accList){
  List<Contact> conList=a.contacts;
  For(Contact c:conList){
    system.debug('Account Name = ' + a.name + ', Contact Name = ' + c.FirstName + ' ' +
c.LastName);
  }
}
```

Enter Apex Code

```
1 ▼ List<Account> accList = [Select Name,
                                (select FirstName, LastName From contacts)
                                From Account];
2
3
4
5 ▼ For(Account a:accList){
6   List<Contact> conList=a.contacts;
7   For(Contact c:conList){
8     system.debug('Account Name = ' + a.name +
9       ', Contact Name = ' + c.FirstName
10      + ' ' + c.LastName);
11   }
12 }
```

Open Log Execute Execute Highlighted

Output:

Developer Console - Google Chrome
btscom-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage

Log executeAnonymous @10/9/2023, 9:00:27 PM

| Execution Log | | |
|---------------|------------|--|
| Timestamp | Event | Details |
| 21:00:27:039 | USER_DEBUG | [8] DEBUG Account Name = Cinemax1, Contact Name = Taemin Kang |
| 21:00:27:075 | USER_DEBUG | [8] DEBUG Account Name = University of Arizona, Contact Name = Jane Grey |
| 21:00:27:075 | USER_DEBUG | [8] DEBUG Account Name = United Oil & Gas Corp., Contact Name = Lauren Boyle |
| 21:00:27:076 | USER_DEBUG | [8] DEBUG Account Name = United Oil & Gas Corp., Contact Name = Stella Pavlova |
| 21:00:27:076 | USER_DEBUG | [8] DEBUG Account Name = United Oil & Gas Corp., Contact Name = Arthur Song |
| 21:00:27:076 | USER_DEBUG | [8] DEBUG Account Name = United Oil & Gas Corp., Contact Name = Avi Green |
| 21:00:27:076 | USER_DEBUG | [8] DEBUG Account Name = sForce, Contact Name = Siddartha Nedaerk |
| 21:00:27:076 | USER_DEBUG | [8] DEBUG Account Name = sForce, Contact Name = Jake Llorrac |
| 21:00:27:076 | USER_DEBUG | [8] DEBUG Account Name = GenePoint, Contact Name = Edna Frank |
| 21:00:27:077 | USER_DEBUG | [8] DEBUG Account Name = United Oil & Gas, UK, Contact Name = Ashley James |
| 21:00:27:077 | USER_DEBUG | [8] DEBUG Account Name = United Oil & Gas, Singapore, Contact Name = Tom Ripley |
| 21:00:27:077 | USER_DEBUG | [8] DEBUG Account Name = United Oil & Gas, Singapore, Contact Name = Liz D'Cruz |
| 21:00:27:077 | USER_DEBUG | [8] DEBUG Account Name = Edge Communications, Contact Name = Rose Gonzalez |
| 21:00:27:077 | USER_DEBUG | [8] DEBUG Account Name = Edge Communications, Contact Name = Sean Forbes |
| 21:00:27:077 | USER_DEBUG | [8] DEBUG Account Name = Burlington Textiles Corp of America, Contact Name = Jack Rogers |
| 21:00:27:078 | USER_DEBUG | [8] DEBUG Account Name = Pyramid Construction Inc., Contact Name = Pat Stumuller |
| 21:00:27:078 | USER_DEBUG | [8] DEBUG Account Name = Dickenson plc, Contact Name = Andy Young |
| 21:00:27:078 | USER_DEBUG | [8] DEBUG Account Name = Grand Hotels & Resorts Ltd, Contact Name = Tim Barr |
| 21:00:27:078 | USER_DEBUG | [8] DEBUG Account Name = Grand Hotels & Resorts Ltd, Contact Name = John Bond |
| 21:00:27:078 | USER_DEBUG | [8] DEBUG Account Name = Express Logistics and Transport, Contact Name = Josh Davis |
| 21:00:27:079 | USER_DEBUG | [8] DEBUG Account Name = Express Logistics and Transport, Contact Name = Babara Levy |
| 21:00:27:079 | USER_DEBUG | [8] DEBUG Account Name = Ramdeobaba, Contact Name = Jihyo Park |



Q26. Write a SOQL query to display 10 Accounts with its Opportunities Names.

Ans. Query Editor: Select Name, (Select Name from Opportunities) from Account Limit 10

Logs Tests Checkpoints **Query Editor** View State Progress Problems

```
Select Name, (Select Name from Opportunities) from Account Limit 10
```

Any query errors will appear here...

Output:

| Query Results - Total Rows: 10 | |
|-------------------------------------|--|
| Name | Opportunities |
| Lee Mihee | |
| Jeon Jungkook | |
| University of Arizona | [{"Name": "University of AZ Portable Generators"}, {"Name": "University of AZ SLA"}, {"Name": "Unive |
| United Oil & Gas Corp. | [{"Name": "United Oil Installations"}, {"Name": "United Oil Installations"}, {"Name": "United Oil Refine |
| sForce | |
| GenePoint | [{"Name": "GenePoint Standby Generator"}, {"Name": "GenePoint Lab Generators"}, {"Name": "GeneP |
| United Oil & Gas, UK | |
| United Oil & Gas, Singapore | |
| Edge Communications | [{"Name": "Edge Emergency Generator"}, {"Name": "Edge Installation"}, {"Name": "Edge Emergency (|
| Burlington Textiles Corp of America | [{"Name": "Burlington Textiles Weaving Plant Generator"}] |

Q27. Write a SOQL query to show 5 Accounts who have SLA value - 'Gold' with its related cases.

Ans. Query Editor:

```
Select Name, SLA__c,(Select CaseNumber, Status from Cases where Status = 'Escalated') From Account where SLA__c='Gold'
```

Output:

| Query Results - Total Rows: 2 | | |
|-------------------------------|--------|---|
| Name | SLA__c | Cases |
| Grand Hotels & Resorts Ltd | Gold | [{"CaseNumber": "00001013", "Status": "Escalated"}] |
| University of Arizona | Gold | |

Q28. Write a SOQL query to display 10 Accounts whose rating is hot with related contact and Opportunity records.

Ans. Query Editor:

```
Select Name, Rating,(Select Name, LeadSource from Contacts),(Select Name, CloseDate from Opportunities) From Account Where Rating = 'Hot' Order By CreatedDate desc Limit 10
```

Output:

| Query Results - Total Rows: 2 | | | |
|-------------------------------|--------|---|--|
| Name | Rating | Contacts | Opportunities |
| United Oil & Gas Corp. | Hot | [{"Name": "Stella Pavlova", "LeadSource": "Public Relations"}, {"Name": "John Doe", "LeadSource": "Email"}] | [{"Name": "United Oil SLA", "CloseDate": "2023-07-07"}, {"Name": "United Oil Office Portable Generator SLA", "CloseDate": "2023-07-07"}] |
| Edge Communications | Hot | [{"Name": "Rose Gonzalez", "LeadSource": "Trade Show"}, {"Name": "David Wilson", "LeadSource": "Email"}] | [{"Name": "Edge SLA", "CloseDate": "2023-04-01"}, {"Name": "Edge Installation", "CloseDate": "2023-04-01"}] |