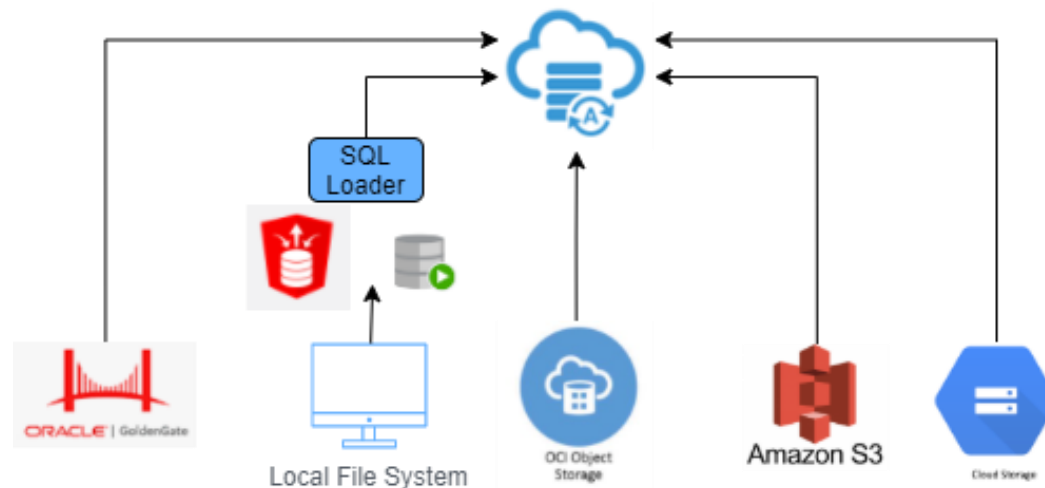


Loading Data into ATP Database

Here are the options to load into Oracle Autonomous Transaction Processing (ATP) Database:

- Load Data from Files in the Cloud (OCI Object Storage/ Amazon S3/ GCP Storage/ Azure BLOB Instance)
 - Copy Data into an Existing Table
 - Load Data Pump Dump File into an Existing Table
- Import Data from OCI Object Storage using Oracle Data Pump
- Load Data from Local Files
 - Using Oracle Database Actions
 - Using Database tools e.g. SQL Developer
 - Using SQL Loader
- Use Oracle GoldenGate to replicate Data with ATP



Loading Data into ATP Database

Load Data from Files in OCI Object Storage

- **Copy Data into an Existing Table**

1. First create object storage credentials and store in the ATP Database. This stores the credentials in the database in an encrypted format.

```
SET DEFINE OFF
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'DEF_CRED_NAME',
    username => 'adb_user@example.com',
    password => 'password'
  );
END;
/
```

- ***credential_name*** name can be any valid name.
- ***username*** should be OCI username to access bucket of Object Storage.
- ***password*** of OCI username.

NOTE:

- “SET DEFINE OFF” is used to disable special characters and allow creating the password properly as SQL Plus treats & as bind parameter
- Credentials needs to be created once only until expires and can be reused for all data loads.
- Creation of credential is not required if resource principal credentials is enabled.

Loading Data into ATP Database

2. Load Data into existing table from a file in Object Storage

```
BEGIN
  DBMS_CLOUD.COPY_DATA(
    table_name => 'CHANNELS',
    credential_name => 'DEF_CRED_NAME',
    file_uri_list => 'https://objectstorage.us-phoenix-1.oraclecloud.com/n/namespace-
string/b/bucketname/o/channels.txt',
    format => json_object('delimiter' value ',')
  );
END;
/
```

- **table_name** is a valid table in ATP Database.
- **credential_name** is the name of the credential created as part of the previous step.
- **file_uri_list** is the source data file name with OCI URL for object storage including namespace and bucket name.
- **format** defines the type of file. In our case it's ',' delimited text file.

Loading Data into ATP Database

Load Data from Files in OCI Object Storage

- Load Data Pump Dump File into an Existing Table

1. First create object storage credentials and store in the ATP Database. This stores the credentials in the database in an encrypted format.

```
SET DEFINE OFF
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'DEF_CRED_NAME',
    username => 'adb_user@example.com',
    password => 'password'
  );
END;
/
```

2. Load Data into existing table from a file in Object Storage

```
BEGIN
  DBMS_CLOUD.COPY_DATA(
    table_name => 'CHANNELS',
    credential_name => 'DEF_CRED_NAME',
    file_uri_list => 'https://objectstorage.us-phoenix-1.oraclecloud.com/n/namespace-
string/b/bucketname/o/exp01.dmp',
    format => json_object('type' value 'datapump')
  );
END;
```

- **file_uri_list** is the source .dmp (exported data dump) file name with OCI URL for object storage including namespace and bucket name.
- **format** defines the type of file. In this case it is datapump file.

Loading Data into ATP Database

Import Data from OCI Object Storage using Oracle Data Pump

1. First, use Oracle Data Pump Export to export all database objects, metadata including data in a .dmp file with Oracle recommended parameters to ensure performance

```
expdp sh/sh@orcl \  
exclude=cluster,indextype,db_link \  
parallel=16 \  
schemas=sh \  
dumpfile=export%u.dmp \  
encryption_pwd_prompt=yes
```

- **schemas** “SH” of source database is exported. So credentials to “SH” schema is provided
 - **exclude** parameter ensure that these objects are not exported. In this case, db_link, clusters, indextypes are excluded.
 - **parallel** is set to 16 which is equal to no. of CPUs available in the database.
 - **encryption_pwd_prompt** is set to yes to encrypt the .dmp with an encryption password (optional)
2. Place the .dmp file to a bucket in Object Storage
 3. Next, use Oracle Data Pump Import to import the exported database objects, metadata including data from .dmp to ATP Database. In order to do so, the following steps need to be followed.
 - a. Create object storage credentials and store in the ATP Database. This stores the credentials in the database in an encrypted format. (Shown before)
 - b. Run Data Pump Import with the dumpfile along with oracle recommended parameters and values and credentials

Loading Data into ATP Database

```
impdp admin/password@db2020adb_high \  
    directory=data_pump_dir \  
    credential=def_cred_name \  
    dumpfile= https://objectstorage.us-ashburn-1.oraclecloud.com/n/namespace-  
string/b/bucketname/o/export%u.dmp \  
    parallel=16 \  
    encryption_pwd_prompt=yes \  
    transform=segment_attributes:n \  
    transform=dwcs_cvt_iots:y transform=constraint_use_default_index:y \  
    exclude=cluster,indextype,db_link
```

- **directory** is the name of the DBA Directory
 - **credential_name** is the name of the credential created as part of the previous step.
 - **dumpfile** is the source .dmp file name with OCI URL for object storage including namespace and bucket name.
 - **parallel** is set to 16 which is equal to no. of OCPUs available in the ATP database.
 - **encryption_pwd_prompt** is set to yes if while exporting the datapumfile this attribute was set to yes.
 - **transform=segment_attributes:n** to allow import of only the basic DDL for all table DDL in the dmp file and omits storage clause, tablespace specification, datafiles parameters and consider the default one set at the ATP Database level.
 - **exclude** parameter ensure that these objects are not exported. In this case, db_link, clusters, indextypes are excluded.
3. Check the Log file for Data Pump Import and places it in the Object Storage. In order to do so, DBMS_CLOUD.PUT_OBJECT PLSQL package is used.

Loading Data into ATP Database

```
BEGIN
  DBMS_CLOUD.PUT_OBJECT(
    credential_name => 'DEF_CRED_NAME',
    object_uri => 'https://objectstorage.us-ashburn-1.oraclecloud.com/n/namespace-
string/b/bucketname/o/import.log',
    directory_name  => 'DATA_PUMP_DIR',
    file_name => 'import.log');
END;
/
```

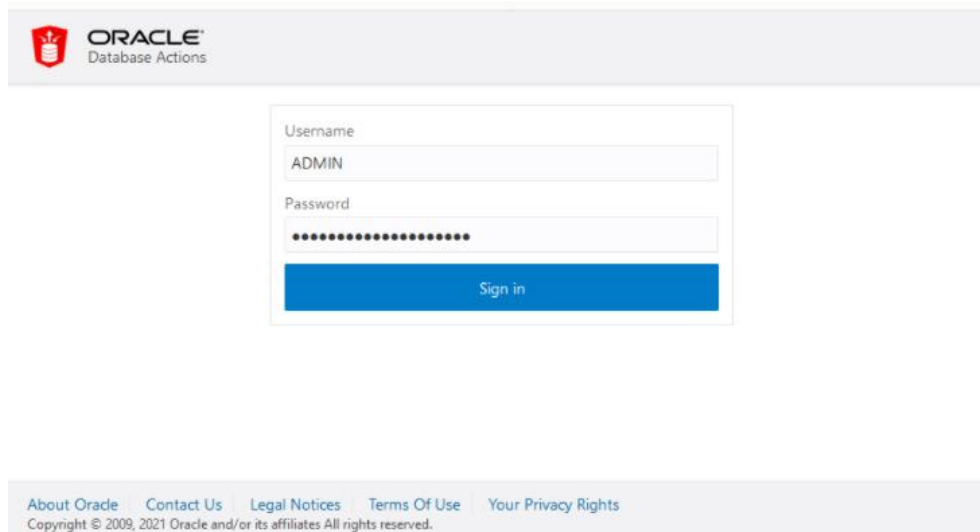
- **credential_name** is the name of the credential created as part of the previous step.
- **object_uri** is the object storage path where the log file will be written to and it looks like OCI URL for object storage including namespace and bucket name.
- **directory_name** is the name of the DBA Directory
- **file_name** is the log file name.

Loading Data into ATP Database

Load Data from Local Files

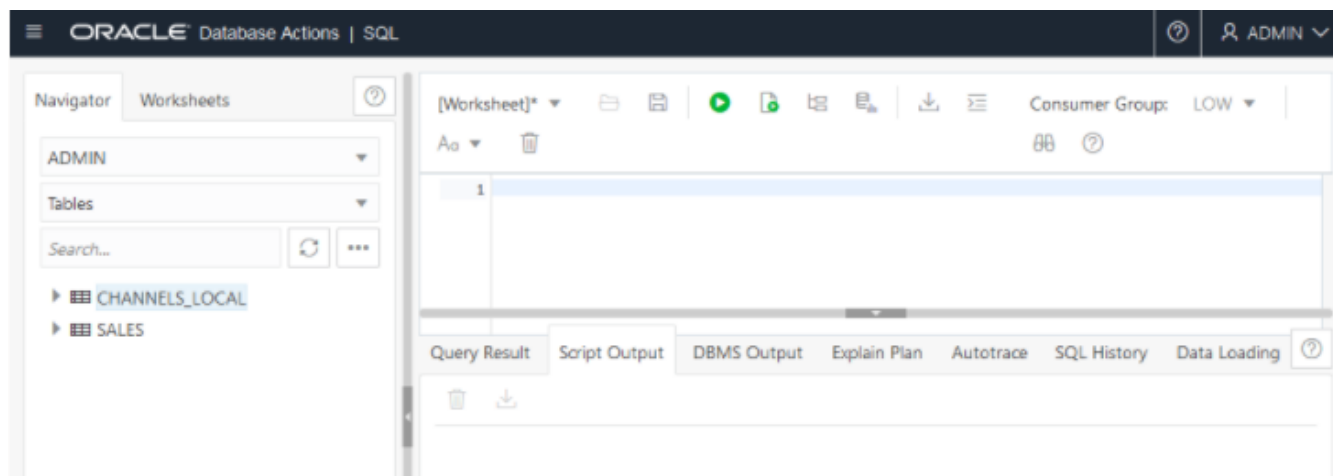
- **Using Oracle Database Actions**

1. Login to Database Actions Page



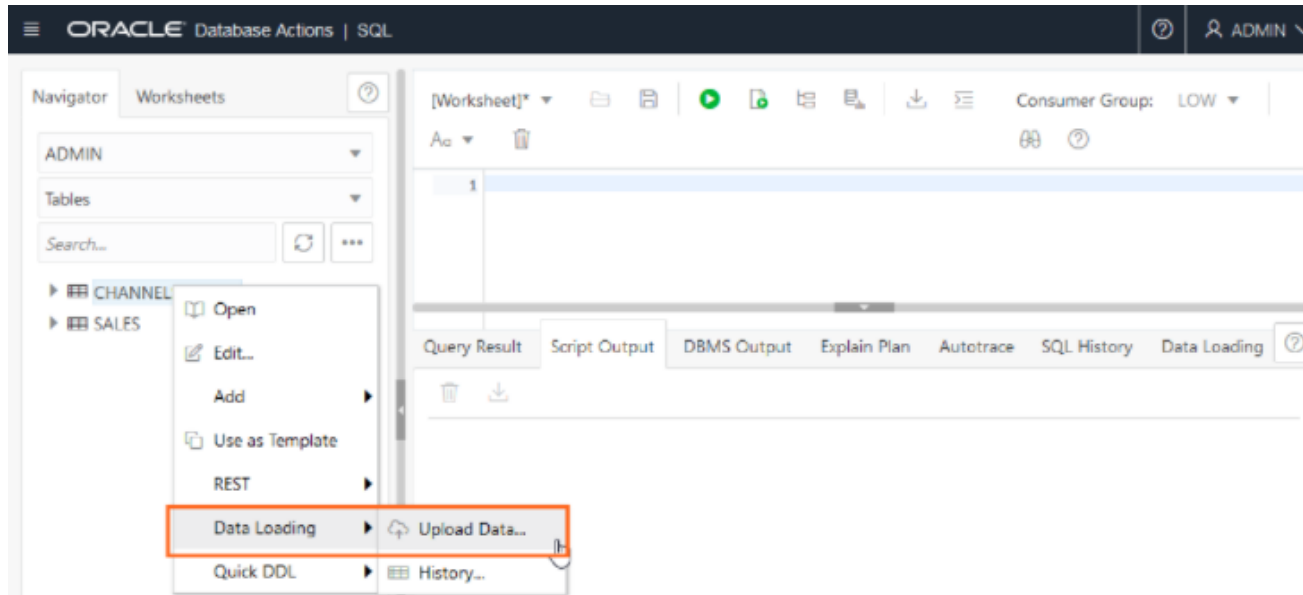
The screenshot shows the Oracle Database Actions login interface. At the top left is the Oracle logo and the text "ORACLE Database Actions". Below this is a login form with two input fields: "Username" containing the text "ADMIN" and "Password" filled with dots. A blue "Sign in" button is positioned below the password field. At the bottom of the page, there is a footer with links: "About Oracle", "Contact Us", "Legal Notices", "Terms Of Use", and "Your Privacy Rights", followed by the copyright notice "Copyright © 2009, 2021 Oracle and/or its affiliates All rights reserved."

2. Navigate to Development, click on SQL

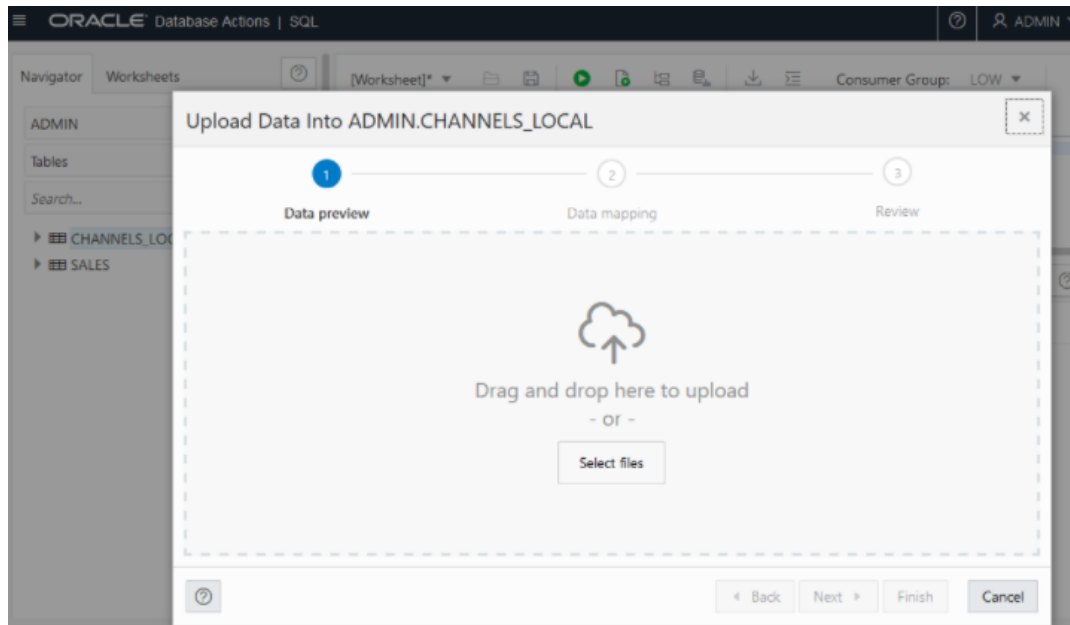


Loading Data into ATP Database

3. In the navigator, choose the table and right click to select Data loading > Upload Data



4. Upload Data wizard launches.

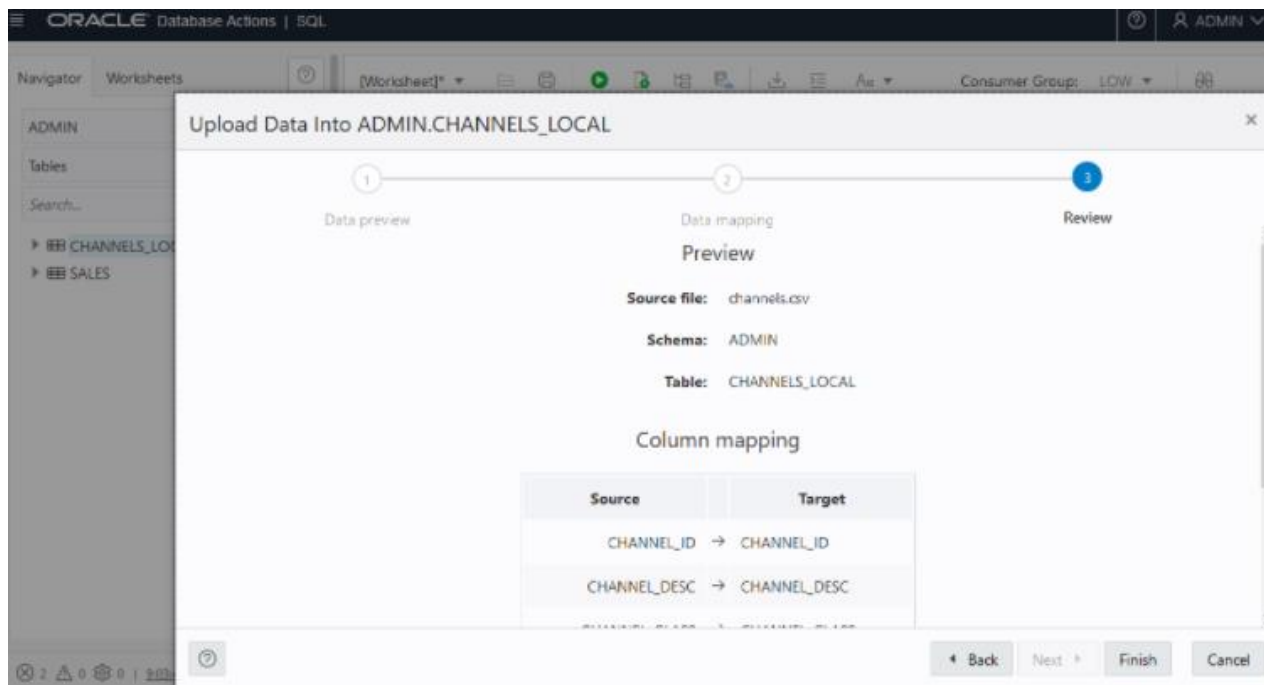


Loading Data into ATP Database

5. In the wizard, either drag and drop or browse the file from local file system. Click Next.
6. Perform the mapping/ formatting between source file columns to target database columns. Click Next.

Note: If there is a problem at this stage, source file needs to be corrected/ fixed before the data import.

7. Review and click Finish.



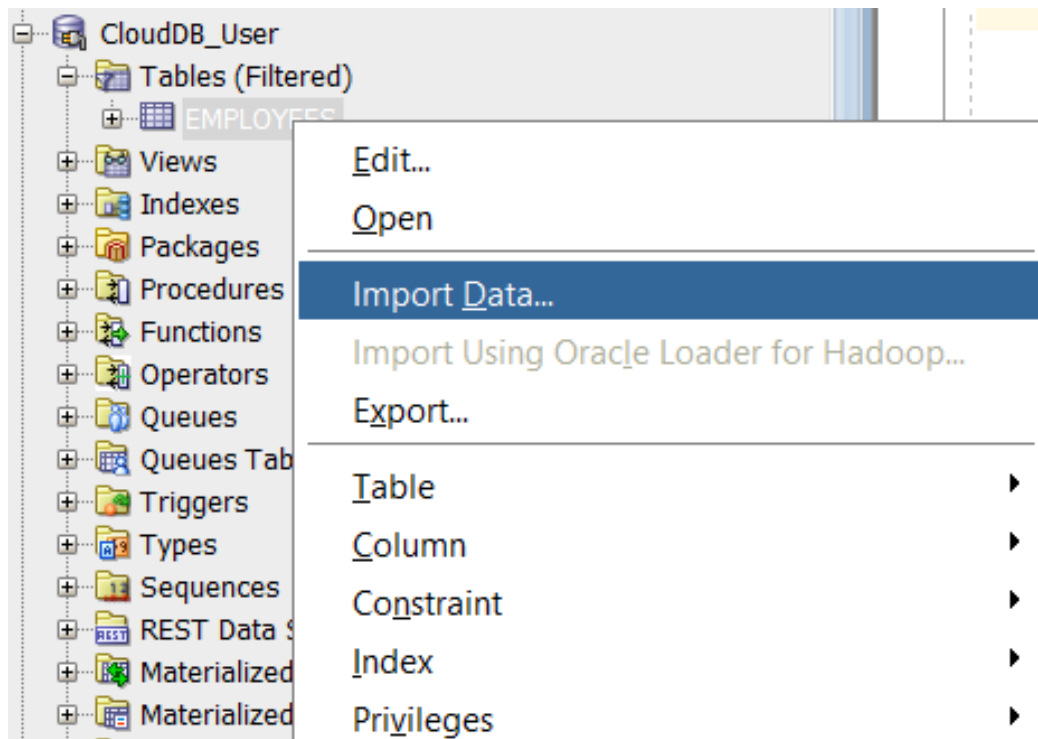
8. Click OK to confirm the import.
9. For detailed Summary of the upload process, right click on the table in the Navigator tab, select Data Loading > select Loaded Data. A summary dialog appears with load summary
10. If any data failed to load, Failed rows column shows the number of rows failed. When clicked on the column, a dialog is displayed with failed rows.

Loading Data into ATP Database

Load Data from Local Files

- **Using Database tools e.g. SQL Developer**

1. Launch SQL Developer and connect to ATP Database
2. Select the Table. Right Click to select Import Data option



3. Wizard launches. Choose the file, change the encoding style and delimiter and other fields (Header, Skip Rows) as required. Click Next

Loading Data into ATP Database

Data Import Wizard - Step 1 of 5

Data Preview

Source: Local File

File: C:\Users\Dell\Desktop\emp_csv.csv

File Format

☒ Header After Skip Skip Rows: 0

Format: csv ☒ Preview Row Limit: 100

Encoding: UTF-8

Delimiter: , Line Terminator: standard: CR LF, CR or LF

Left Enclosure: " Right Enclosure: "

File Contents

empid	empname
1	Raja
2	Leena

Help < Back Next > Finish Cancel

4. Select Import Method. Click Next.

Data Import Wizard - Step 2 of 4

Import Method

Specify the method for importing data. For insert and sends it to a worksheet.

Import Method: Insert

Table Name:

☐ Import Row Limit: 100

File Contents

empid	empname
1	Raja
2	Leena

Loading Data into ATP Database

5. Choose Columns. Click Next.

Data Import Wizard - Step 3 of 5

Choose Columns

Select the columns to import from the data set and arrange them in the order you want.

Available Columns

Selected Columns

empid
empname

File Contents

empid	empname
1	Raja
2	Leena

Help < Back Next > Finish Cancel

6. Perform Mapping. Click Next

Data Import Wizard - Step 4 of 5

Column Definition

For each column in the Source Data Columns list on the left, select a Target Table column on the right.

Match By Name

Source Data Columns

empid
empname

Target Table Columns

Name NAME
Data Type VARCHAR2
Size/Precision 100
☒ Nullable? Default
Comment

Data

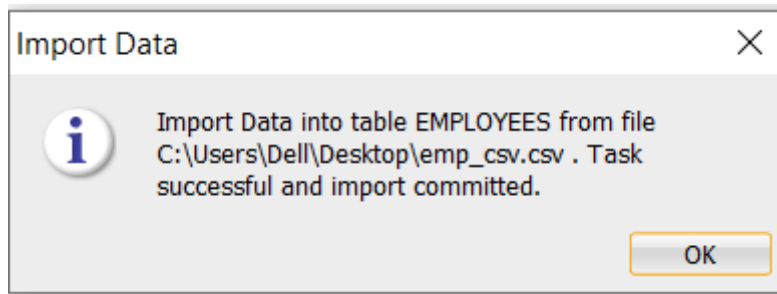
Raja
Leena

Status

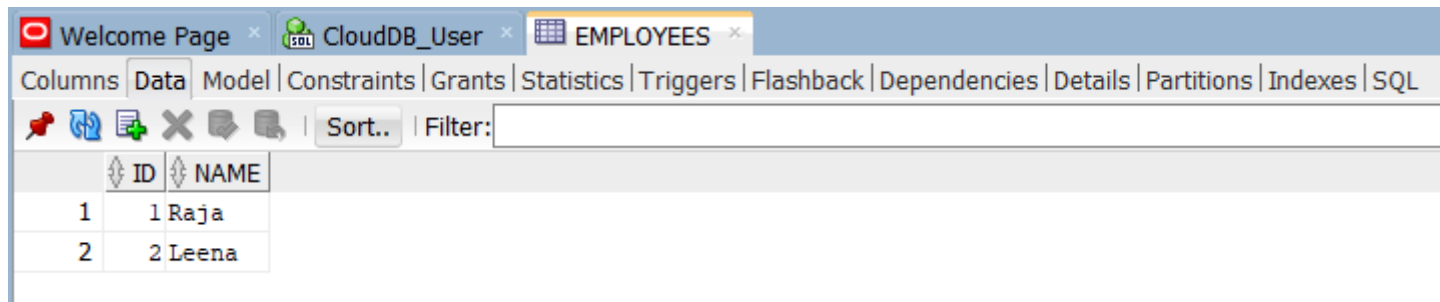
Help < Back Next > Finish Cancel

Loading Data into ATP Database

7. Review and Click Finish.
8. Once imported, dialog appears as shown below.



9. Click OK
10. Click on Table and verify the data.

A screenshot of a database management interface. The top bar shows three tabs: "Welcome Page", "CloudDB_User", and "EMPLOYEES". Below the tabs is a menu bar with options: "Columns", "Data", "Model", "Constraints", "Grants", "Statistics", "Triggers", "Flashback", "Dependencies", "Details", "Partitions", "Indexes", and "SQL". Below the menu bar is a toolbar with icons for various actions, including a "Sort.." button and a "Filter:" input field. The main area displays a table with two columns: "ID" and "NAME". The table contains two rows of data: "1 Raja" and "2 Leena".

ID	NAME
1	1 Raja
2	2 Leena

11. Or, a simple select statement can be written to verify as well.

Loading Data into ATP Database

• Using SQL Loader

1. Data will come in Input Data Files from the source system
2. Loader Control file contains DDL instructions that SQL Loader uses to determine where to find data, how to parse and interpret and where to insert
3. SQL Loader utility is run to load data into Database table
4. Log file is generated with all load related details
5. The rejected records are pushed to Bad File
6. The records which doesn't meet the selection criteria is pushed to Discard file
7. Data is inserted into the DB Table with indexes rebuild

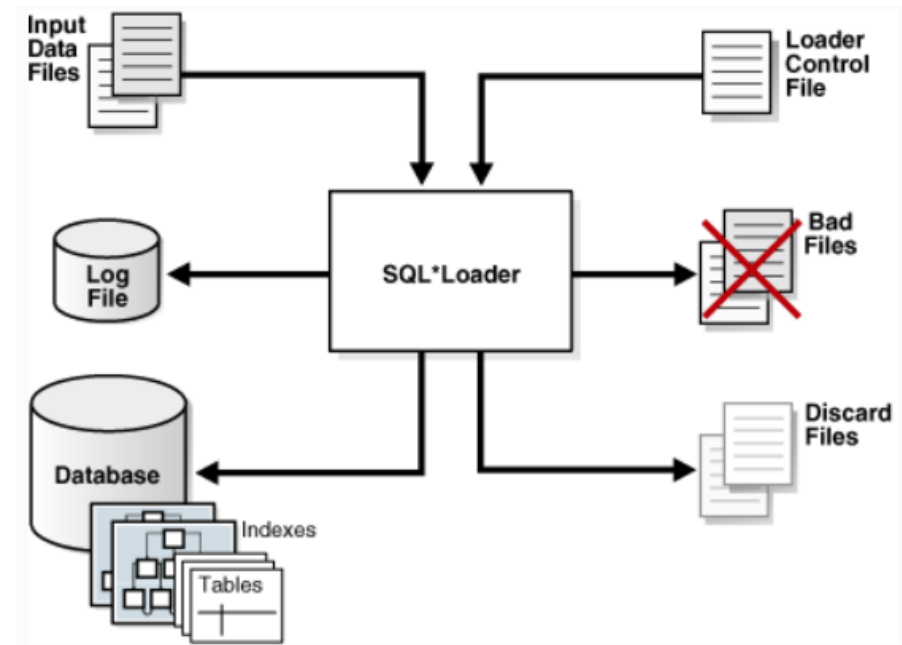
NOTE

- Suitable for small amounts of data as the load performance depends on the network bandwidth between client and ATP Database
- For large amount of data, SQL Loader is not recommended since client will have gather schema stats by themselves if Oracle recommended parameters are not used.
- Oracle recommends SQL Loader parameters as mentioned below for best performance:

`readsize=100M`

`bindsize=100M`

`direct=N`



Loading Data into ATP Database

Use Oracle GoldenGate to replicate Data with ATP

- **Replication for different use cases:** Report offloading, Active-Active, Cloud-Cloud and Cloud-On-premise
- **Inter-region and cross region replication:** Replicate data between different Oracle Cloud Data centers across globe.
- **Replicate between targets:** Replicate from ATP Database to any other target database that GoldenGate supports