Employee.csv

```
EMPLOYEE_ID,NAME,SALARY,DEPARTMENT_ID,JOINING_DATE
100,'Jennifer',4400,10,'2017-01-05'
101,'Michael',13000,10,'2018-08-24'
102,'Pat',6000,10,'2018-12-10'
103,'Den', 11000,20,'2019-02-17'
104,'Alexander',3100,20,'2019-07-01'
105,'Shelli',2900,20,'2020-04-22'
106,'Sigal',2800,30,'2020-09-05'
107,'Guy',2600,30,'2021-05-25'
108,'Karen',2500,30,'2021-12-21'
```

1. Create an Employee table in database **bootcamp** and in schema **snow.**
    a. Requirements :
        i. Create a table with name as Employee on Employee.csv
        ii. Copy data from snowflake Employee table to a output S3 dir.
    b. Expectations - I want you to provide / submit followings
        i. Commands to create table and load data.
        ii. Commands to copy data from snowflake table to s3 dir

Answer:

Code used in Snowflake:

1. **Set up compute warehouse**
   I have created a small compute warehouse `aws_warehouse` with size `x-small` to run my queries and process data efficiently.

```
create warehouse aws_warehouse with warehouse_size="x-small";
```

2. **Created database and schema**
   I have created a database `bootcamp` and a schema `bootcamp.snow` to organize my tables.

```
create database bootcamp;
```

create schema bootcamp.snow;

3. **Created Employee table**
   I have created a table `Employee` with columns `EMPLOYEE_ID`, `NAME`, `SALARY`, `DEPARTMENT_ID`, and `JOINING_DATE` to store employee data from `Employee.csv`.

```
CREATE OR REPLACE TABLE Employee (
  EMPLOYEE_ID INT,
  NAME STRING,
  SALARY INT,
  DEPARTMENT_ID INT,
  JOINING_DATE DATE
);
```

4. **Inserted data into Employee table**
   I have inserted sample employee data into the table to test it.

```
insert into employee values
(100, 'Jennifer', 4400, 10, '2017-01-05'),
(101, 'Michael', 13000, 10, '2018-08-24'),
(102, 'Pat', 6000, 10, '2018-12-10'),
(103, 'Den', 11000, 20, '2019-02-17'),
(104, 'Alexander', 3100, 20, '2019-07-01'),
(105, 'Shelli', 2900, 20, '2020-04-22'),
(106, 'Sigal', 2800, 30, '2020-09-05'),
(107, 'Guy', 2600, 30, '2021-05-25'),
(108, 'Karen', 2500, 30, '2021-12-21')
;
```

5. **Verified the data**
   I have run a `SELECT` query to make sure the data is correctly loaded.

```
select * from employee;
```

6. **Configured Snowflake to connect to S3**
   I have created a **storage integration** so Snowflake can securely connect to my AWS S3 bucket using an IAM role.

```
Create or replace storage integration
  Snow_OBJSS
  type = external_stage
  storage_provider = s3
  enabled = true
  storage_aws_role_arn =
  'arn:aws:iam::930797749854:role/snowflakerolenew'
  STORAGE_ALLOWED_LOCATIONS = ('s3://amzn-snowflake-s3-bucket/')
  STORAGE_AWS_EXTERNAL_ID =
'SWA67785_SFCRole=3_P7v6KQ4q9qmSKORmx8rS03v/mxo=';
```

I have verified that the integration works using:

```
desc integration Snow_OBJSS;
```

7. **Created file format for CSV**
   I have defined a CSV file format so Snowflake can export data as a CSV file with headers and no compression.

```
create or replace file format csv_formatt
  type = csv field_delimiter = ',' skip_header = 1 null_if = ('NULL','null')
```

```
  empty_field_as_null = TRUE
  COMPRESSION = NONE;
```

8. **Created stage pointing to S3**
   I have set up a stage `s3_stage_snowflake` that points to my S3 bucket using the storage integration and file format.

```
create or replace stage s3_stage_snowflake
  storage_integration = Snow_OBJSS
  url = 's3://amzn-snowflake-s3-bucket/'
  file_format = csv_formatt;
```

9. **Exported Employee table to S3**
   I have copied the Employee table data to S3 as a single CSV file with column headers.

```
COPY INTO @s3_stage_snowflake/data.csv
FROM employee
FILE_FORMAT = csv_formatt
SINGLE = TRUE
OVERWRITE = TRUE
HEADER = TRUE;
```

10. **AWS setup**
    I have created an IAM role `snowflakerolenew` with two policies:

    snowflakepolicynew (managed)

snowflakes3 (inline policy to allow S3 read/write/list)

Inline policy code:

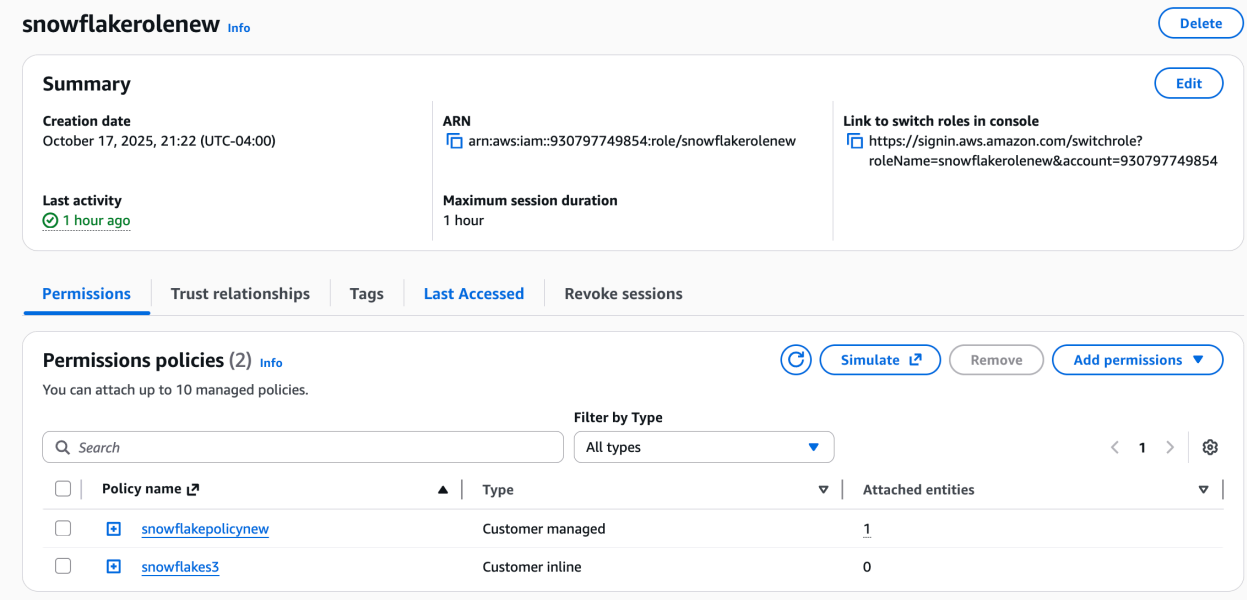
```
{
	"Version": "2012-10-17",
	"Statement": [
		{
			"Effect": "Allow",
			"Action": [
				"s3:PutObject",
				"s3:GetObject",
				"s3:ListBucket"
			],
			"Resource": [
				"arn:aws:s3:::amzn-snowflake-s3-bucket",
				"arn:aws:s3:::amzn-snowflake-s3-bucket/*"
			]
		}
	]
}
```

This policy gives Snowflake the **exact permissions it needs** to **write CSV files to S3**, **read them if needed**, and **list files in the bucket**.

11. **Set trust relationship**
    I have set the trust relationship so Snowflake can assume the IAM role securely with an ExternalId.

```
{
  "Version": "2012-10-17",
```

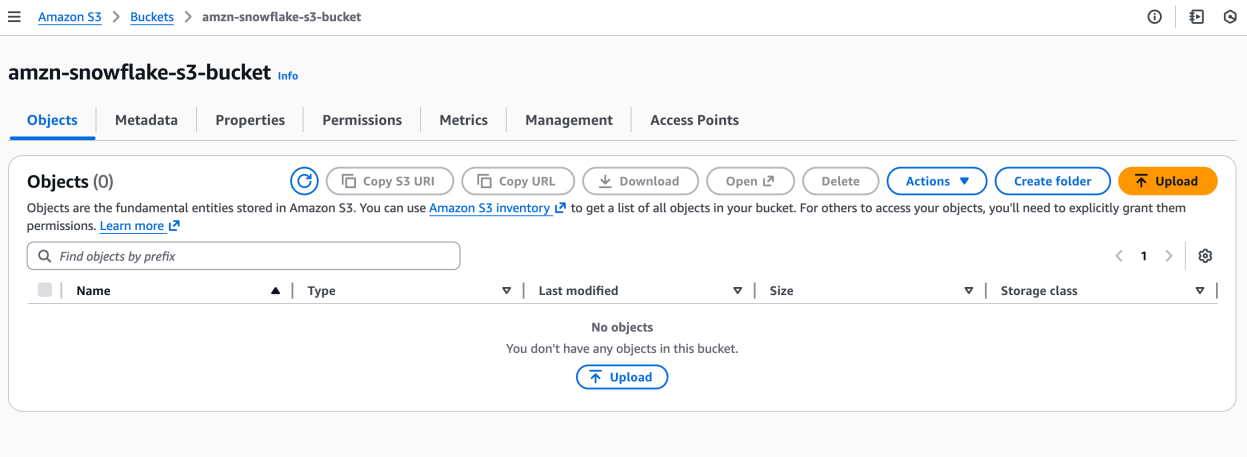
```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::974916068036:user/externalstages/cisx7c0000"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "SWA67785_SFCRole=3_P7v6KQ4q9qmSKORmx8rS03v/mxo="
        }
      }
    }
  ]
}
```



12. Created S3 bucket

I have created an S3 bucket `amzn-snowflake-s3-bucket` to store the exported CSV file. After the `COPY INTO` command, the file appeared in the bucket successfully.

- Before file is copied



- After file is copied