

Homework 9

1. Create a cluster named 'redshift_hw9' (only use free tier)

The screenshot shows the AWS Amazon Redshift console. At the top, there is a green success message: "redshift-hw9 has been successfully created." Below it, a blue banner says "Try new Amazon Redshift features in preview. Create a cluster with preview features. Production use of the cluster is not supported. Use this cluster for testing only." On the left, the navigation bar shows "Amazon Redshift > Clusters". Under "In my account", there is a section titled "Connect to Redshift clusters" with three tabs: "Query data using Redshift query editor", "Work with your client tools", and "Choose your JDBC or ODBC driver". The "Work with your client tools" tab shows a dropdown for "Cluster" set to "redshift-hw9", and buttons for "Copy JDBC URL" and "Copy ODBC URL". The "Choose your JDBC or ODBC driver" tab shows a dropdown for "Driver" set to "JDBC 4.2 without AWS SDK (.jar)". Below these sections is a table titled "Clusters (1) info" with one row. The row contains columns for Cluster (redshift-hw9), Status (Available), Cluster namespace (b4e649f8-1baf-4258-...), Availability Zone (us-east-2a), Multi-AZ (No), Storage capacity us... (160 GB), and CPU utilization (N/A). The table has filters at the top and pagination controls (1 of 1).

Cluster	Status	Cluster namespace	Availability Zone	Multi-AZ	Storage capacity us...	CPU utilization
redshift-hw9 dc2.large 1 node 160 GB	Available	b4e649f8-1baf-4258-...	us-east-2a	No	160 GB	N/A

2. Create>Show the required IAM roles for the task.

AWS Services Search [Option+S] Global nikhilgudur

Identity and Access Management (IAM)

Role Redshift created.

IAM > Roles

Roles (6) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search Role name Trusted entities

Role name	Trusted entities
AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Role)
AWSServiceRoleForRedshift	AWS Service: redshift (Service-Linked Role)
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)
rds-monitoring-role	AWS Service: monitoring.rds
Redshift	AWS Service: redshift

Roles Anywhere Info

Authenticate your non AWS workloads and securely provide access to AWS services.

Manage

Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization methods as your AWS services.

X.509 Standard

Use your own existing PKI infrastructure or use [AWS Certificate Manager Private Certificate](#).

Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

The screenshot shows the AWS Redshift console for managing a cluster named `vpc-0acc664c5072dec86`. The cluster is located in the `us-east-2a` region. Key configuration details include:

- Subnet group:** `default` (Enhanced VPC routing: `Disabled`)
- Endpoint URL:** `-`
- IP address type:** `-`
- Allow connections from outside the VPC:** `Disabled`

Cluster permissions:

A callout box provides instructions: "Create an IAM role as the default for this cluster that has the [AmazonRedshiftAllCommandsFullAccess](#) policy attached. This policy includes permissions to run SQL commands to COPY, UNLOAD, and query data with Amazon Redshift. The policy also grants permissions to run SELECT statements for related services, such as Amazon S3, Amazon CloudWatch logs, Amazon SageMaker, and AWS Glue."

Associated IAM roles (1) [Info](#)

One IAM role is associated: `Redshift` (Status: `in-sync`, Role type: `--`). Buttons for `Set default` and `Manage IAM roles` are available.

Granted accounts (0)

No accounts have been granted access. Buttons for `Edit`, `Revoke`, and `Grant access` are present.

3. Create an S3 bucket and load the tickitdb files.

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with the AWS logo, a 'Services' dropdown, a search bar containing 'Search [Option+S]', and user information for 'nikhilgudur'. Below the navigation bar, a green banner displays a success message: 'Upload succeeded' with a link to 'View details below.' On the left, a sidebar shows a 'VPC' icon. The main content area has a title 'Upload: status' with a 'Close' button. A note at the top says: 'The information below will no longer be available after you navigate away from this page.' Below this, a 'Summary' section shows destination details: 'Destination s3://s3red.hw' with 'Succeeded' status and '7 files, 27.9 MB (100.00%)', and 'Failed' status with '0 files, 0 B (0%)'. There are tabs for 'Files and folders' (which is selected) and 'Configuration'. The 'Files and folders' tab displays a table titled 'Files and folders (7 Total, 27.9 MB)'. The table includes a search bar 'Find by name' and columns: Name, Folder, Type, Size, Status, and Error. The data in the table is as follows:

Name	Folder	Type	Size	Status	Error
allevents_pipe.txt	tickitdb/	text/plain	435.4 KB	SUCCEEDED	-
allusers_pipe.txt	tickitdb/	text/plain	5.6 MB	SUCCEEDED	-
listings_pipe.txt	tickitdb/	text/plain	11.0 MB	SUCCEEDED	-
date2008_pipe.txt	tickitdb/	text/plain	14.2 KB	SUCCEEDED	-
venue_pipe.txt	tickitdb/	text/plain	7.8 KB	SUCCEEDED	-

4. Create tables on your cluster. (CATEGORY table, DATE table, EVENT table, VENUE table, USERS table, LISTING table, SALES table)

```
CREATE TABLE demoaws.public.CATEGORY
(cat_id INT PRIMARY KEY,
cat_name VARCHAR(255),
cat_desc VARCHAR(255));
```

The screenshot shows the Amazon Redshift Query editor interface. The top navigation bar includes the AWS logo, Services, a search bar, and user information (Ohio, nikhilgudur). The main area has tabs for Editor, Query history, Saved queries, and Scheduled queries. The Editor tab is selected, showing a status bar with 'Connected' to database 'demoaws' as user 'awsuser'. A sidebar on the left displays 'Resources info' with sections for Select databaseinfo (showing 'demoaws' selected) and Select schemainfo (showing 'public' selected). The main query editor window contains a query titled 'Query 1' with the following SQL code:

```
1 CREATE TABLE demoaws.public.CATEGORY
2   (cat_id INT PRIMARY KEY,
3    cat_name VARCHAR(255),
4    cat_desc VARCHAR(255));
```

Below the query are buttons for Run, Save, Schedule, and Clear. The 'Query results' section indicates the query completed successfully on November 13, 2023, at 19:55:40, with an elapsed time of 00 m 02 s. There are also tabs for Table details, Execution, Data, and Visualize.

```
CREATE TABLE demoaws.public.DATE (
    date_id INT PRIMARY KEY,
    event_date DATE,
    day_of_week VARCHAR(20),
    week_number INT,
    month_name VARCHAR(20),
    quarter_name VARCHAR(20),
    year INT,
    is_holiday BOOLEAN
);
```

The screenshot shows the Amazon Redshift Query Editor interface. On the left, there's a sidebar titled 'Resources info' with dropdown menus for 'Select databaseinfo' (set to 'demoaws') and 'Select schemainfo' (set to 'public'). Below these are sections for 'category_pkey', 'date_pkey', 'category', and 'date'. The main area is titled 'Query 1' and contains the following SQL code:

```
1 CREATE TABLE demoaws.public.DATE (
2     date_id INT PRIMARY KEY,
3     event_date DATE,
4     day_of_week VARCHAR(20),
5     week_number INT,
6     month_name VARCHAR(20),
7     quarter_name VARCHAR(20),
8     year INT,
9     is_holiday BOOLEAN
10 );
```

Below the code are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. To the right, there's a status bar showing 'Connected' to 'demoaws' with user 'awsuser'. At the bottom, there are tabs for 'Query results' and 'Table details', along with buttons for 'Execution', 'Data', and 'Visualize'. The status bar also includes links for 'CloudShell', 'Feedback', and copyright information.

```
CREATE TABLE demoaws.public.EVENT_DETAILS (
    event_id INT PRIMARY KEY,
    venue_id INT,
    cat_id INT,
    date_id INT,
    event_title VARCHAR(255),
    start_time TIMESTAMP
);
```

The screenshot shows the Amazon Redshift Query editor interface. On the left, there's a sidebar titled 'Resources info' with dropdown menus for 'Select databaseinfo' (set to 'demoaws') and 'Select schemainfo' (set to 'public'). Below these are lists of tables: 'category_pkey', 'date_pkey', 'event_details_pkey', 'category', 'date', and 'event_details'. The main area is titled 'Query 1' and contains the following SQL code:

```
1 CREATE TABLE demoaws.public.EVENT_DETAILS (
2     event_id INT PRIMARY KEY,
3     venue_id INT,
4     cat_id INT,
5     date_id INT,
6     event_title VARCHAR(255),
7     start_time TIMESTAMP
8 );
```

Below the code are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. To the right of the code area are various connection and session settings. At the bottom, there are tabs for 'Query results' and 'Table details', along with buttons for 'Execution', 'Data', and 'Visualize'. The status bar at the bottom indicates the query was completed on November 13, 2023, at 20:06:48, with an elapsed time of 00 m 04 s.

```
CREATE TABLE demoaws.public.LOCATION (
    loc_id INT PRIMARY KEY,
    loc_name VARCHAR(255),
    loc_city VARCHAR(255),
    loc_state VARCHAR(20),
    available_seats INT
);
```

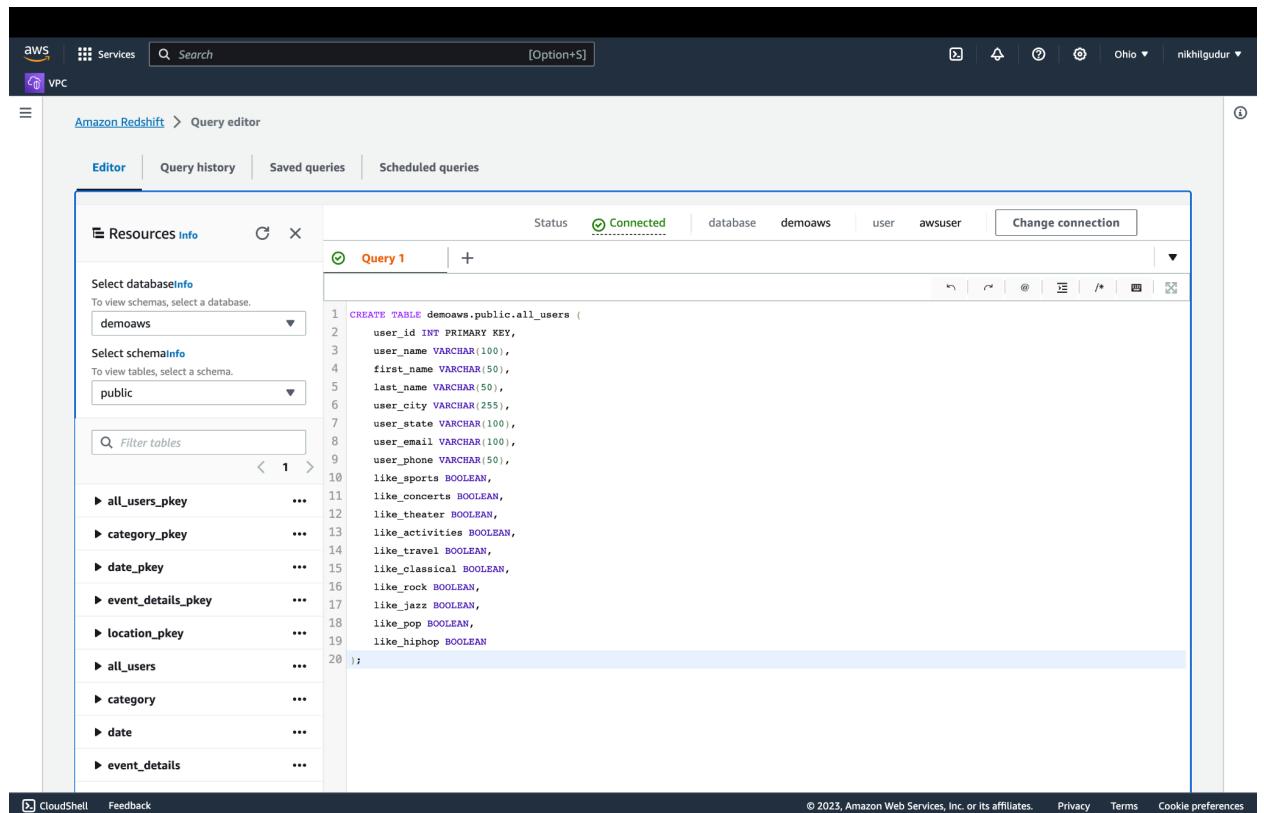
The screenshot shows the Amazon Redshift Query editor interface. On the left, there's a sidebar titled 'Resources info' with dropdown menus for 'Select databaseInfo' (set to 'demoaws') and 'Select schemaInfo' (set to 'public'). Below these are lists of tables: 'category_pkey', 'date_pkey', 'event_details_pkey', 'location_pkey', 'category', 'date', 'event_details', and 'location'. The main area is titled 'Query 1' and contains the following SQL code:

```
1 CREATE TABLE demoaws.public.LOCATION (
2     loc_id INT PRIMARY KEY,
3     loc_name VARCHAR(255),
4     loc_city VARCHAR(255),
5     loc_state VARCHAR(20),
6     available_seats INT
7 );
8
```

Below the code are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. To the right, there's a status bar showing 'Connected' to 'demoaws' with user 'awsuser'. At the bottom, tabs for 'Query results' and 'Table details' are visible, along with buttons for 'Execution', 'Data', and 'Visualize'. A message at the bottom indicates the query completed successfully on November 13, 2023, at 20:11:33, with an elapsed time of 00 m 07 s.

```
CREATE TABLE demoaws.public.all_users (
    user_id INT PRIMARY KEY,
    user_name VARCHAR(100),
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    user_city VARCHAR(255),
    user_state VARCHAR(100),
    user_email VARCHAR(100),
    user_phone VARCHAR(50),
    like_sports BOOLEAN,
    like_concerts BOOLEAN,
    like_theater BOOLEAN,
    like_activities BOOLEAN,
    like_travel BOOLEAN,
    like_classical BOOLEAN,
    like_rock BOOLEAN,
    like_jazz BOOLEAN,
    like_pop BOOLEAN,
    like_hiphop BOOLEAN
```

```
);
```



The screenshot shows the AWS Amazon Redshift Query Editor interface. The top navigation bar includes the AWS logo, Services, Search, and user information (Ohio, nikhilgudur). Below the navigation is a toolbar with various icons. The main area has tabs for Editor, Query history, Saved queries, and Scheduled queries. The Editor tab is active, showing a query titled "Query 1". The query code is as follows:

```
1 CREATE TABLE demoaws.public.all_users (
2     user_id INT PRIMARY KEY,
3     user_name VARCHAR(100),
4     first_name VARCHAR(50),
5     last_name VARCHAR(50),
6     user_city VARCHAR(255),
7     user_state VARCHAR(100),
8     user_email VARCHAR(100),
9     user_phone VARCHAR(50),
10    like_sports BOOLEAN,
11    like_concerts BOOLEAN,
12    like_theater BOOLEAN,
13    like_activities BOOLEAN,
14    like_travel BOOLEAN,
15    like_classical BOOLEAN,
16    like_rock BOOLEAN,
17    like_jazz BOOLEAN,
18    like_pop BOOLEAN,
19    like_hiphop BOOLEAN
20 );
```

The left sidebar contains "Resources info" sections for "Select databaseInfo" (demoaws) and "Select schemaInfo" (public). It also features a "Filter tables" search bar and a table list with entries like all_users_pkey, category_pkey, date_pkey, event_details_pkey, location_pkey, all_users, category, date, and event_details, each with a "..." button.

```
CREATE TABLE demoaws.public.LISTINGS (
    listing_id INT PRIMARY KEY,
    sell_user_id INT,
    associated_event_id INT,
    date_number SMALLINT,
    number_of_tickets SMALLINT,
    price_per_ticket DECIMAL(10,3),
    price DECIMAL(10,3),
    listing_timestamp TIMESTAMP
);
```

The screenshot shows the Amazon Redshift Query editor interface. The top navigation bar includes the AWS logo, Services, Search, and a user dropdown for nikhilgudur. The main area has tabs for Editor, Query history, Saved queries, and Scheduled queries, with Editor selected. A sidebar titled 'Resources info' shows the database 'demoaws' and schema 'public'. The central workspace displays a query titled 'Query 1' with the following SQL code:

```
1 CREATE TABLE demoaws.public.LISTINGS (
2     listing_id INT PRIMARY KEY,
3     sell_user_id INT,
4     associated_event_id INT,
5     date_number SMALLINT,
6     number_of_tickets SMALLINT,
7     price_per_ticket DECIMAL(10,3),
8     price DECIMAL(10,3),
9     listing_timestamp TIMESTAMP
10 );
11
```

Below the code are buttons for Run, Save, Schedule, and Clear. The status bar indicates the query completed successfully on November 13, 2023, at 20:20:55, with an elapsed time of 00 m 02 s. There are also tabs for Query results and Table details, along with Execution, Data, and Visualize buttons.

```
CREATE TABLE demoaws.public.SALES (
    sales_id INT PRIMARY KEY,
    listing_id INT,
    seller_user_id INT,
    buyer_user_id INT,
    associated_event_id INT,
    date_number SMALLINT,
    quantity_sold SMALLINT,
    price_paid DECIMAL(10,3),
    commission DECIMAL(10,3),
    sale_timestamp TIMESTAMP
);
```

The screenshot shows the Amazon Redshift Query Editor interface. At the top, there's a navigation bar with 'aws' and 'Services' tabs, a search bar, and user information ('nikhilgudur'). Below the navigation is a breadcrumb trail: 'Amazon Redshift > Query editor'. The main area has tabs for 'Editor' (which is selected), 'Query history', 'Saved queries', and 'Scheduled queries'. On the left, a sidebar titled 'Resources info' lists database and schema selection dropdowns ('demoaws' and 'public'), a 'Filter tables' search bar, and a list of tables: 'all_users_pkey', 'category_pkey', 'date_pkey', 'event_details_pkey', 'listings_pkey', 'location_pkey', 'sales_pkey', 'all_users', and 'category'. The central panel shows a query editor with a status bar indicating 'Connected' to 'demoaws' as 'user' 'awouser'. A green status bar at the top of the editor says 'Query 1'. The code area contains the following SQL:

```
1 CREATE TABLE demoaws.public.SALES (
2     sales_id INT PRIMARY KEY,
3     listing_id INT,
4     seller_user_id INT,
5     buyer_user_id INT,
6     associated_event_id INT,
7     date_number SMALLINT,
8     quantity_sold SMALLINT,
9     price_paid DECIMAL(10,3),
10    commission DECIMAL(10,3),
11    sale_timestamp TIMESTAMP
12 );
13
```

Below the code are buttons for 'Run', 'Save', 'Schedule', and 'Clear'. To the right is a 'Send feedback' link. The bottom section is titled 'Query results' and shows a message: 'Completed, started on November 13, 2023 at 20:25:47' and 'ELAPSED TIME: 03 m 52 s'. There are also 'Execution', 'Data', and 'Visualize' buttons.

5. Copy the data from S3 to Redshift cluster 'redshift_hw9'.

```
COPY demoaws.public.DATE
FROM 's3://s3red.hw/ticketdb/date2008_pipe.txt'
IAM_ROLE 'arn:aws:iam::785133169911:role/Redshift'
DELIMITER '|'
CSV;
SELECT * FROM demoaws.public.DATE;
```

The screenshot shows the AWS Redshift Data Explorer interface. On the left, there's a sidebar with a tree view of tables: event_details_pkey, listings_pkey, location_pkey, sales_pkey, all_users, category, date, event_details, listings, location, and sales. The 'date' node is expanded, showing its sub-columns: date_id, event_date, day_of_week, week_number, month_name, quarter_name, year, and is_holiday. The main area displays a table titled 'Query 3577' with 365 rows returned. The table has columns: date_id, event_date, day_of_week, week_number, month_name, quarter_name, year, and is_holiday. The data shows various dates from January 1, 2008, to January 10, 2008, with some entries marked as holidays. At the bottom, there are links for CloudShell, Feedback, and a footer with copyright information.

date_id	event_date	day_of_week	week_number	month_name	quarter_name	year	is_holiday
1827	2008-01-01	WE	1	JAN	1	2008	true
1828	2008-01-02	TH	1	JAN	1	2008	false
1829	2008-01-03	FR	1	JAN	1	2008	false
1830	2008-01-04	SA	2	JAN	1	2008	false
1831	2008-01-05	SU	2	JAN	1	2008	false
1832	2008-01-06	MO	2	JAN	1	2008	false
1833	2008-01-07	TU	2	JAN	1	2008	false
1834	2008-01-08	WE	2	JAN	1	2008	false
1835	2008-01-09	TH	2	JAN	1	2008	false
1836	2008-01-10	FR	2	JAN	1	2008	false

```
COPY demoaws.public.event_details
FROM 's3://s3red.hw/ticketdb/allevents_pipe.txt'
IAM_ROLE 'arn:aws:iam::785133169911:role/Redshift'
DELIMITER '|'
CSV;
```

The screenshot shows the AWS Redshift Data Explorer interface. On the left, there's a sidebar with a tree view of tables: event_details_pkey, listings_pkey, location_pkey, sales_pkey, all_users, category, date (with sub-items date_id, event_date, day_of_week, week_number, month_name, quarter_name, year, is_holiday), event_details, listings, location, and sales. The date node is expanded. In the main area, the 'Query results' tab is selected, showing a table titled 'Query 3693'. The status bar indicates it was completed on November 13, 2023, at 20:56:55, with an elapsed time of 00 m 13 s. Below this, a table titled 'Rows returned (8798)' displays 10 rows of data from the event_details table. The columns are event_id, venue_id, cat_id, date_id, event_title, and start_time. The data includes various operas and their details.

event_id	venue_id	cat_id	date_id	event_title	start_time
1	305	8	1851	Gotterdammerung	2008-01-25 14:30:00
2	306	8	2114	Boris Godunov	2008-10-15 20:00:00
3	302	8	1935	Salome	2008-04-19 14:30:00
4	309	8	2090	La Cenerentola (Cinderella)	2008-09-21 14:30:00
5	302	8	1982	Il Trovatore	2008-06-05 19:00:00
6	308	8	2109	L'Elisir d'Amore	2008-10-10 19:30:00
7	309	8	1891	Doctor Atomic	2008-03-06 14:00:00
8	302	8	1832	The Magic Flute	2008-01-06 20:00:00
9	308	8	2087	The Fly	2008-09-18 19:30:00
10	305	8	2079	Rigoletto	2008-09-10 15:00:00

```
COPY demoaws.public.location
FROM 's3://s3red.hw/ticketdb/venue_pipe.txt'
IAM_ROLE 'arn:aws:iam::785133169911:role/Redshift'
DELIMITER '|'
CSV;
```

The screenshot shows the AWS Redshift console interface. At the top, there's a navigation bar with the AWS logo, a 'Services' dropdown, a search bar, and user information ('nikhilgudur'). Below the navigation is a sidebar containing a tree view of database tables: 'listings_pkey', 'location_pkey', 'sales_pkey', 'all_users', 'category', 'date', 'event_details', 'listings', 'location' (expanded to show 'loc_id', 'loc_name', 'loc_city', 'loc_state', 'available_seats'), and 'sales'. The main area is titled 'Query results' and shows a completed query (ID 3775) from November 13, 2023, at 20:58:41, with an elapsed time of 00 m 07 s. It displays 202 rows returned. The data is presented in a table with columns: loc_id, loc_name, loc_city, loc_state, and available_seats. The table contains 10 rows of stadium data, such as Toyota Park in Bridgeview, IL, and Gillette Stadium in Foxborough, MA.

loc_id	loc_name	loc_city	loc_state	available_seats
1	Toyota Park	Bridgeview	IL	0
2	Columbus Crew Stadium	Columbus	OH	0
3	RFK Stadium	Washington	DC	0
4	CommunityAmerica Ballpark	Kansas City	KS	0
5	Gillette Stadium	Foxborough	MA	68756
6	New York Giants Stadium	East Rutherford	NJ	80242
7	BMO Field	Toronto	ON	0
8	The Home Depot Center	Carson	CA	0
9	Dick's Sporting Goods Park	Commerce City	CO	0
10	Pizza Hut Park	Frisco	TX	0

```
COPY demoaws.public.all_users
FROM 's3://s3red.hw/ticketdb/allusers_pipe.txt'
IAM_ROLE 'arn:aws:iam::785133169911:role/Redshift'
DELIMITER '|'
CSV;
```

aws | Services | Search [Option+S] | Execution | Data | Visualize | ⓘ

VPC | Query 3863 | Completed, started on November 13, 2023 at 21:00:55 | ELAPSED TIME: 00 m 23 s | Rows returned (49990) | Export | ⓘ

sales_pkey
all_users
user_id
user_name
first_name
last_name
user_city
user_state
user_email
user_phone
like_sports
like_concerts
like_theater
like_activities
like_travel
like_classical
like_rock
like_jazz
like_pop
like_hiphop
category
date
event_details
listings
location
sales

user_id	user_name	first_name	last_name	user_city	user_state	user_email
2	PGL08LJI	Vladimir	Humphrey	Murfreesboro	SK	Suspendisse.tristique@nonnisAenean.edu
4	XDZ38RDD	Barry	Roy	Omaha	AB	sed@iacusUtneec.ca
5	AEB55QTM	Reagan	Hodge	Forest Lake	NS	Cum@accumsan.com
7	OWY35QYB	Tamekah	Juarez	Moultrie	WV	elementum@semperpretiumneque.ca
9	MSD36KVR	Mufutau	Watkins	Port Orford	MD	Integer.mollis.Integer@tristiquealiquet.org
10	WKW41AIW	Naida	Calderon	Waterbury	MB	Donec.fringilla@sodalesat.org
15	OWU78MTR	Scarlett	Mayer	Gadsden	GA	lorem.ipsum@Vestibulumante.com
16	ZMG93CDD	Kieran	Drake	Hot Springs	BC	molestie.tellus@dapibusgravidaAliquam.co
18	VDP05MXU	Germaine	Valdez	Kokomo	WY	cursus.Integer@arcuVestibulumante.com
19	CXQ97IWP	Amal	Landry	Lomita	NT	euismod@turpis.org

CloudShell | Feedback | © 2023, Amazon Web Services, Inc. or its affiliates. | Privacy | Terms | Cookie preferences

```
COPY demoaws.public.event_details
FROM 's3://s3red.hw/ticketdb/allevents_pipe.txt'
IAM_ROLE 'arn:aws:iam::785133169911:role/Redshift'
DELIMITER '|'
CSV;
```

The screenshot shows the AWS Redshift Data Explorer interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, a search bar, and user information ('nikhilgudur'). Below the navigation is a sidebar with a tree view of tables: 'location_pkey', 'sales_pkey', 'all_users', 'category', 'date', 'event_details' (expanded to show 'event_id', 'venue_id', 'cat_id', 'date_id', 'event_title', 'start_time'), 'listings', 'location', and 'sales'. The main area displays a query result titled 'Query 3984' which completed successfully on November 13, 2023 at 21:03:39 with an elapsed time of 00 m 03 s. The results are shown in a table titled 'Rows returned (17596)' with an 'Export' button. The table has columns: event_id, venue_id, cat_id, date_id, event_title, start_time. The data shows various events like 'Gotterdamerung' and 'Salome' occurring between 2008-01-25 and 2008-09-19.

event_id	venue_id	cat_id	date_id	event_title	start_time
1	305	8	1851	Gotterdamerung	2008-01-25 14:30:00
3	302	8	1935	Salome	2008-04-19 14:30:00
5	302	8	1982	Il Trovatore	2008-06-05 19:00:00
7	309	8	1891	Doctor Atomic	2008-03-06 14:00:00
9	308	8	2087	The Fly	2008-09-18 19:30:00
11	302	8	1952	Doctor Atomic	2008-05-01 19:30:00
13	301	8	2154	Lucia di Lammermoor	2008-11-24 15:00:00
15	307	8	2038	Die Walkure	2008-07-31 19:30:00
17	301	8	1834	La Gioconda	2008-01-08 19:00:00
19	302	8	2088	Boris Godunov	2008-09-19 14:30:00

```
COPY demoaws.public.SALES
FROM 's3://s3red.hw/ticketdb/sales_tab.txt'
IAM_ROLE 'arn:aws:iam::785133169911:role/Redshift'
DELIMITER '\t'
TIMEFORMAT 'MM/DD/YYYY HH:MI:SS';
```

The screenshot shows the AWS Redshift Query Editor interface. The top navigation bar includes 'aws', 'Services', 'Search', '[Option+S]', and user information 'nikhilgudur'. Below the navigation is a sidebar with database and table navigation links like 'location_pkey', 'sales_pkey', 'all_users', 'category', 'date', 'event_details', 'listings', 'location', and 'sales'. The main area displays the results of a query with the following details:

- Completed:** started on November 13, 2023 at 21:20:46
- ELAPSED TIME:** 00 m 38 s
- Rows returned:** 158602
- Export:** button
- Search rows:** input field
- Pagination:** 1 to 15861

The results table has the following columns and data:

sales_id	listing_id	seller_user_id	buyer_user_id	associated_event_id	date_number	quantity_sold
2	4	8117	11498	4337	1983	2
4	5	1616	19715	8647	1986	1
5	6	47402	14115	8240	2069	2
7	10	24858	7952	3375	2003	4
9	10	24858	29891	3375	2029	3
10	12	45635	10542	4769	2044	1
15	15	46833	19715	7910	1970	1
16	17	20228	36376	7837	1908	2
18	20	6300	23690	2075	1904	1
19	20	6300	32200	2075	1914	4

6. Write a query in the Redshift query editor to find the top 3 sellers in **San Diego, New York, Boston, and San Jose** based on the number of tickets sold in 2008. (Use Window functions)

```
WITH Question6 AS (
  SELECT
    u.user_city AS CITY,
    u.user_state AS STATE,
    u.user_id AS USERID,
    u.user_name AS USERNAME,
    SUM(s.quantity_sold) AS TotalTicketsSold,
    RANK() OVER (PARTITION BY u.user_city, u.user_state ORDER BY
    SUM(s.quantity_sold) DESC) AS SellerRank
  FROM
    demoaws.public.SALES s
  JOIN
    demoaws.public.all_users u ON s.seller_user_id = u.user_id
  JOIN
```

```

demoaws.public.DATE d ON s.date_number = d.date_id
WHERE
    EXTRACT(YEAR FROM d.event_date) = 2008
GROUP BY
    u.user_city,
    u.user_state,
    u.user_id,
    u.user_name
)
SELECT
    CITY,
    STATE,
    USERID,
    USERNAME,
    TotalTicketsSold
FROM
    Question6
WHERE
    SellerRank <= 3;

```

The screenshot shows the AWS CloudShell interface with the following details:

- Query ID:** Query 5544
- Status:** Completed, started on November 13, 2023 at 22:04:02. ELAPSED TIME: 00 m 16 s
- Rows returned:** 42069
- Execution Tab:** Active tab.
- Data Tab:** Shows a table with columns: city, state, userid, username, and totalticketssold.
- Visualize Tab:** Available but not selected.

city	state	userid	username	totalticketssold
Aberdeen	AZ	2629	BGW64YUW	7
Aberdeen	CA	13927	UDT43NWN	12
Aberdeen	CA	24517	TZF85ZJX	3
Aberdeen	HI	46520	BJG04DJR	17
Aberdeen	HI	34667	ZFL62ORF	9
Aberdeen	IL	29300	ZIK22IR	10
Aberdeen	KY	223	JSN53AUD	1
Aberdeen	MN	8665	LHO40NRQ	4
Aberdeen	NC	19644	HYW35INN	11
Aberdeen	NE	950	BKR90PRR	9

7. Write a query in the Redshift query editor to see events with the lowest sales.

```
SELECT
    ed.event_title AS EVENTNAME,
    COUNT(sa.sales_id) AS TotalSales
FROM
    demoaws.public.EVENT_DETAILS ed
LEFT JOIN
    demoaws.public.SALES sa ON ed.event_id = sa.associated_event_id
GROUP BY
    ed.event_title
ORDER BY
    TotalSales
LIMIT 5;
```

The screenshot shows the AWS Redshift Query Editor interface. The top navigation bar includes the AWS logo, Services, a search bar, and user information (Ohio, nikhilgudur). The main area displays a table of query results.

Table Structure:

Category_pkey	...
date_pkey	...
event_details_pkey	...
listings_pkey	...
location_pkey	...
sales_pkey	...
all_users	...
category	...
date	...
event_details	...
listings	...
location	...
sales	...
sales_id	
listing_id	
seller_user_id	
buyer_user_id	
associated_event_id	
date_number	
quantity_sold	
price_paid	
commission	
sale_timestamp	

Query Results:

eventname	totalsales
White Christmas	20
Joshua Radin	38
Martina McBride	50
Linda Ronstadt	56
Indigo Girls	57

8. Write a query in the Redshift query editor to count the number of users in each state and city combination.

```
SELECT
    user_city AS CITY,
    user_state AS STATE,
    COUNT(user_id) AS TotalUsers
FROM
    demoaws.public.all_users
GROUP BY
    user_city, user_state;
```

The screenshot shows the AWS CloudShell interface with the following details:

- Services:** Services menu is open, showing options like AWS Lambda, Amazon S3, and Amazon Redshift.
- Search:** Search bar at the top right.
- User:** nikhilgudur
- Region:** Ohio
- Query Results:** The main pane displays the results of the executed query. The table has columns: city, state, and totalusers. The data is as follows:

city	state	totalusers
Forest Lake	NS	3
Moultrie	WV	1
Port Orford	MD	1
Waterbury	MB	2
Gadsden	GA	4
Hot Springs	BC	4
Kokomo	WY	1
Walnut	WI	1
Fort Worth	MB	3
Buffalo	CT	1

- Execution:** Execution status: Completed, started on November 13, 2023 at 22:07:53, Elapsed time: 00 m 10 s.
- Data:** Data visualization options.
- Visualize:** Visualize button.
- Export:** Export button.
- Table Details:** Shows the schema of the table used in the query.
- CloudShell:** CloudShell button.
- Feedback:** Feedback button.
- Footer:** © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences