CS443
Cloud Computing
Homework 3 Report

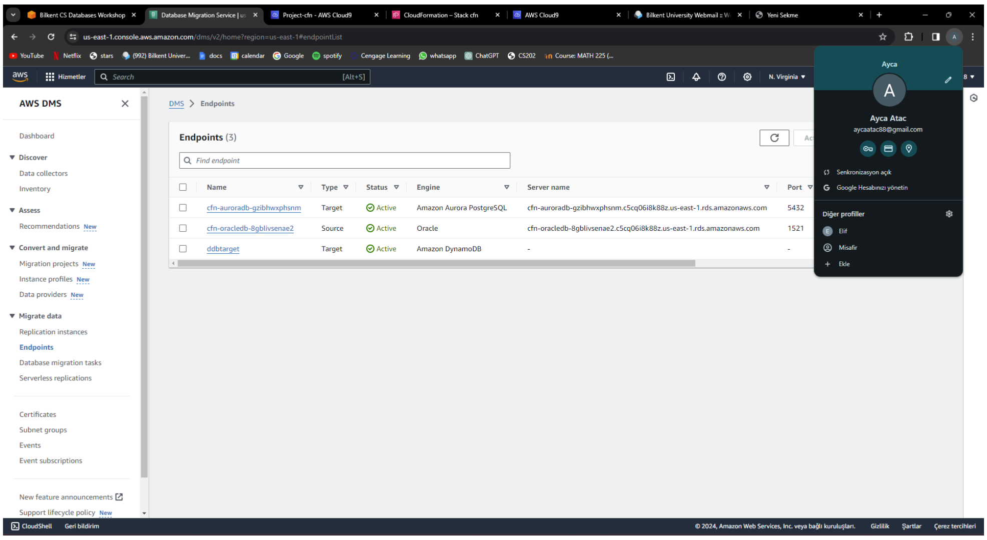
Onurcan Ataç
22002194

Date: 07.04.2024

I have explained the work I have done for each part of the lab separately in this report. I also added related screenshots to the parts. In the account with my Bilkent mail, there was a bug about the Oracle source endpoint, I was getting a *"Test Endpoint failed: Application-Status: 1020912, Application-Message: Invalid username or password."* error even though I was entering the same password that is successfully accepted by the Oracle SQLPLUS login. Therefore I have switched computers and accounts to complete the homework. I used the email and computer of my sister in order to get rid of the bug. When I applied the same steps on the new account, the bug did not emerge. **Therefore, the name on my screenshots is *Ayça Ataç*, the name of my sister since the account and computer were under her name. I still completed all parts of the homework myself.** In the AWS console, my username was automatically "WSParticipantRole/Participant", therefore I used her Google account popup to prove I was the one doing the work in the screenshots.

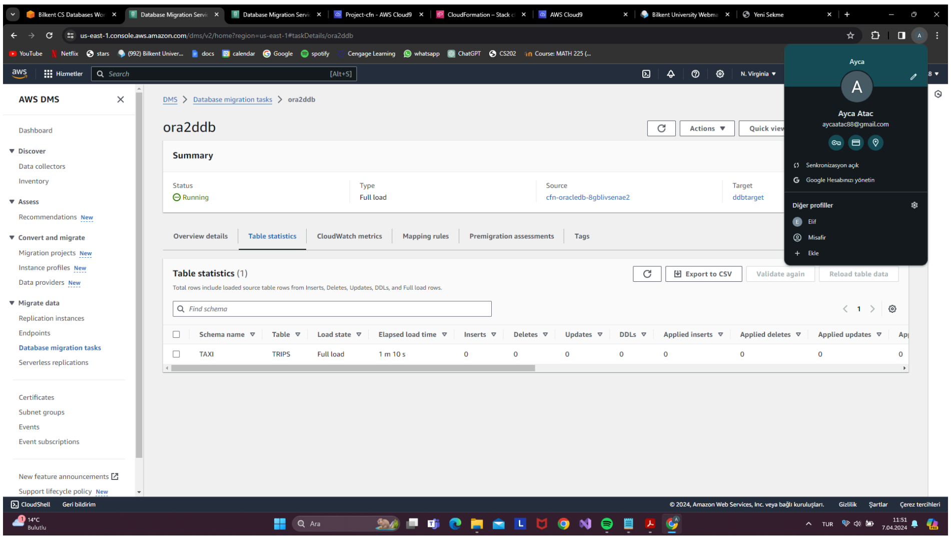
1. Lab1 - Data Migration using AWS DMS

In the first part of the assignment, I performed a migration using AWS DMS from RDS Oracle to Amazon DynamoDB and Amazon Aurora PostgreSQL databases. I used AWS Cloud 9 IDE to connect to Oracle SQLPLUS and PostgreSQL, and other coding purposes throughout the labs. I then created a taxi schema and verified the objects in the schema in Oracle. I also created the schema in PostgreSQL. I then created the source and target endpoints and tested their connections. After getting the successful results from the tests, I created replication tasks and monitored the migrations. I validated the results to finish up this part.



Endpoints (3)

	Name	Type	Status	Engine	Server name	Port
□	cfn-auroradb-gzibhwxphsnm	Target	✓ Active	Amazon Aurora PostgreSQL	cfn-auroradb-gzibhwxphsnm.c5cq06i8k88z.us-east-1.rds.amazonaws.com	5432
□	cfn-oracledb-8gblivsenae2	Source	✓ Active	Oracle	cfn-oracledb-8gblivsenae2.c5cq06i8k88z.us-east-1.rds.amazonaws.com	1521
□	ddbtarget	Target	✓ Active	Amazon DynamoDB	-	-



us-east-1.console.aws.amazon.com/dms/v2/home?region=us-east-1#taskDetails/ora2ddb

DMS > Database migration tasks > ora2ddb

ora2ddb

Summary

Status	Type	Source	Target
Running	Full load	cfn-oracledb-8gblivsenae2	ddbtarget

Overview details | Table statistics | CloudWatch metrics | Mapping rules | Premigration assessments | Tags

Table statistics (1)

Export to CSV | Validate again | Reload table data

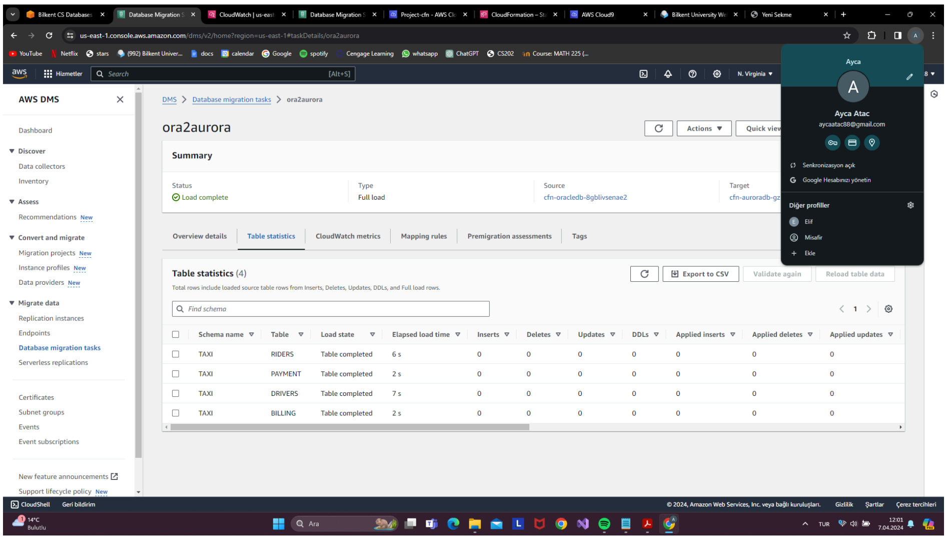
Total rows include loaded source table rows from Inserts, Deletes, Updates, DDLs, and Full load rows.

Find schema

Schema name	Table	Load state	Elapsed load time	Inserts	Deletes	Updates	DDLs	Applied inserts	Applied deletes	Applied updates	Ap
TAXI	TRIPS	Full load	1 m 10 s	0	0	0	0	0	0	0	0

© 2024, Amazon Web Services, Inc. veya bağlı kuruluşları. Gizlilik Şartlar Çerez tercihleri

11:53 7.04.2024



us-east-1.console.aws.amazon.com/dms/v2/home?region=us-east-1#taskDetails/ora2aurora

DMS > Database migration tasks > ora2aurora

ora2aurora

Summary

Status	Type	Source	Target
Load complete	Full load	cfn-oracledb-8gblivsenae2	cfn-auroradb-gz

Overview details | Table statistics | CloudWatch metrics | Mapping rules | Premigration assessments | Tags

Table statistics (4)

Export to CSV | Validate again | Reload table data

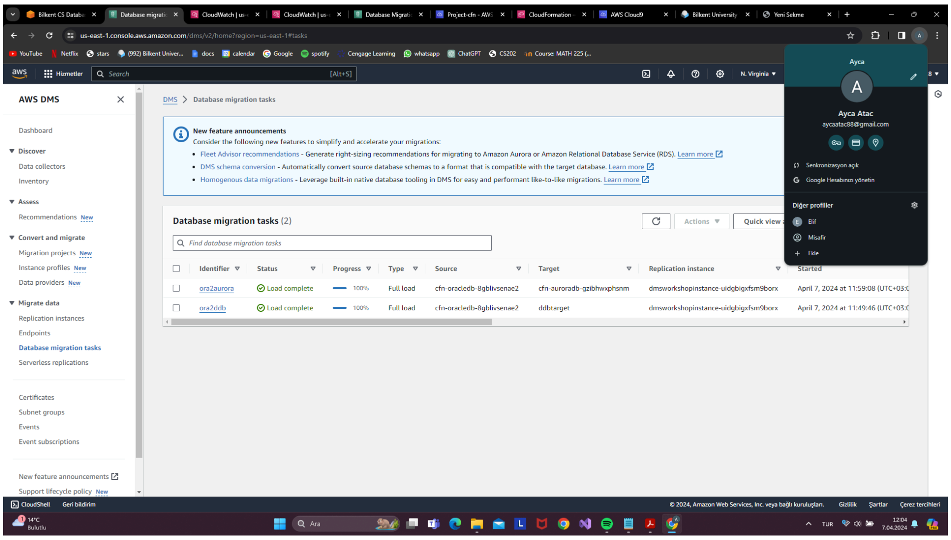
Total rows include loaded source table rows from Inserts, Deletes, Updates, DDLs, and Full load rows.

Find schema

Schema name	Table	Load state	Elapsed load time	Inserts	Deletes	Updates	DDLs	Applied inserts	Applied deletes	Applied updates
TAXI	RIDERS	Table completed	6 s	0	0	0	0	0	0	0
TAXI	PAYMENT	Table completed	2 s	0	0	0	0	0	0	0
TAXI	DRIVERS	Table completed	7 s	0	0	0	0	0	0	0
TAXI	BILLING	Table completed	2 s	0	0	0	0	0	0	0

© 2024, Amazon Web Services, Inc. veya bağlı kuruluşları. Gizlilik Şartlar Çerez tercihleri

12:01 7.04.2024



us-east-1.console.aws.amazon.com/dms/v2/home?region=us-east-1#tasks

DMS > Database migration tasks

New feature announcements

Consider the following new features to simplify and accelerate your migrations:

- Fleet Advisor recommendations - Generate right-sizing recommendations for migrating to Amazon Aurora or Amazon Relational Database Service (RDS). Learn more
- DMS schema conversion - Automatically convert source database schemas to a format that is compatible with the target database. Learn more
- Homogenous data migrations - Leverage built-in native database tooling in DMS for easy and performant like-to-like migrations. Learn more

Database migration tasks (2)

Find database migration tasks

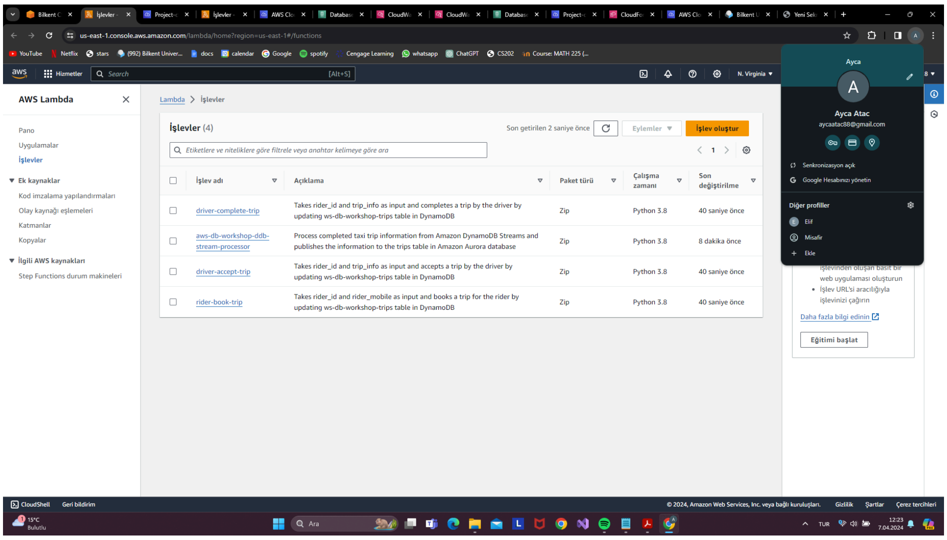
Identifier	Status	Progress	Type	Source	Target	Replication instance	Started
ora2aurora	Load complete	100%	Full load	cfn-oracledb-8gblivsenae2	cfn-auroradb-gzibhwxphsnm	dmsworkshopinstance-uidgbigxfsm9borx	April 7, 2024 at 11:59:08 (UTC+03:0
ora2ddb	Load complete	100%	Full load	cfn-oracledb-8gblivsenae2	ddbtarget	dmsworkshopinstance-uidgbigxfsm9borx	April 7, 2024 at 11:49:46 (UTC+03:0

© 2024, Amazon Web Services, Inc. veya bağlı kuruluşları. Gizlilik Şartlar Çerez tercihleri

12:04 7.04.2024

2. Lab2 - Data processing using Amazon DynamoDB and Amazon Aurora

I continued with the second part in which I created and used an API to add functionality to the taxi data and pass data from Amazon DynamoDB to the Aurora database using Lambda functions. I started by enabling streams for the DynamoDB tables. I continued with creating and deploying AWS Lambda functions. I then created a REST API by using Amazon API Gateway, by connecting the resources with corresponding Lambda functions. I then called the API functions via specific HTTP requests and therefore simulated the taxi ride workflow.



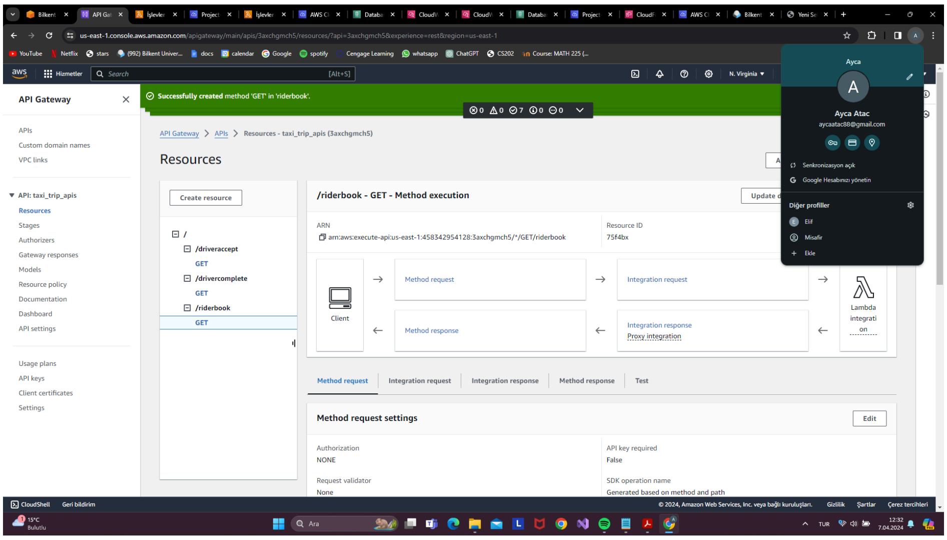
us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions

Lambda > İşlevler

İşlevler (4)

Son getirilen 2 saniye önce

İşlev adı	Açıklama	Paket türü	Çalışma zamanı	Son değiştirilme
driver-complete-trip	Takes rider_id and trip_info as input and completes a trip by the driver by updating ws-db-workshop-trips table in DynamoDB	Zip	Python 3.8	40 saniye önce
aws-db-workshop-ddb-stream-processor	Process completed taxi trip information from Amazon DynamoDB Streams and publishes the information to the trips table in Amazon Aurora database	Zip	Python 3.8	8 dakika önce
driver-accept-trip	Takes rider_id and trip_info as input and accepts a trip by the driver by updating ws-db-workshop-trips table in DynamoDB	Zip	Python 3.8	40 saniye önce
rider-book-trip	Takes rider_id and rider_mobile as input and books a trip for the rider by updating ws-db-workshop-trips table in DynamoDB	Zip	Python 3.8	40 saniye önce



us-east-1.console.aws.amazon.com/apigateway/main/apis/3axchgmch5/resources/?api=3axchgmch5&experience=rest&region=us-east-1

Successfully created method 'GET' in 'riderbook'.

API Gateway > APIs > Resources - taxi_trip_apis (3axchgmch5)

Resources

/
- /driveraccept
  - GET
- /drivercomplete
  - GET
- /riderbook
  - GET

/riderbook - GET - Method execution

ARN: arn:aws:execute-api:us-east-1:458342954128:3axchgmch5/*/GET/riderbook

Resource ID: 75f4bx

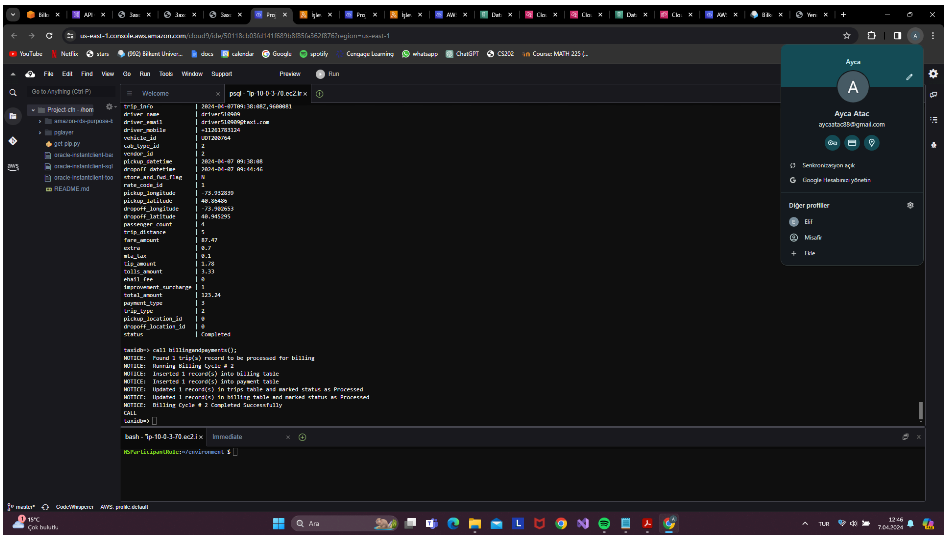
Method request settings

Authorization: NONE

API key required: False

Request validator: None

SDK operation name: Generated based on method and path



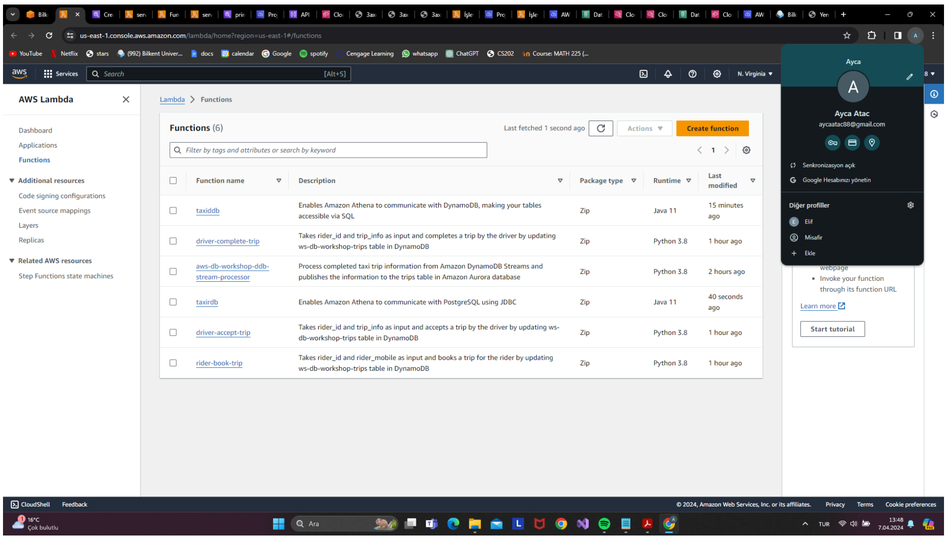
```
trip_info             | 2024-04-07T09:38:08Z,9600081
driver_name           | driver510909
driver_email          | driver510909@taxi.com
driver_mobile         | +11261783124
vehicle_id            | UDT200764
cab_type_id           | 2
vendor_id             | 2
pickup_datetime       | 2024-04-07 09:38:08
dropoff_datetime      | 2024-04-07 09:44:46
store_and_fwd_flag    | N
rate_code_id          | 1
pickup_longitude      | -73.932839
pickup_latitude       | 40.86486
dropoff_longitude     | -73.902653
dropoff_latitude      | 40.945295
passenger_count       | 4
trip_distance         | 5
fare_amount           | 87.47
extra                 | 0.7
mta_tax               | 0.1
tip_amount            | 1.78
tolls_amount          | 3.33
ehail_fee             | 0
improvement_surcharge | 1
total_amount          | 123.24
payment_type          | 3
trip_type             | 2
pickup_location_id    | 0
dropoff_location_id   | 0
status                | Completed

taxidb=> call billingandpayments();
NOTICE:  Found 1 trip(s) record to be processed for billing
NOTICE:  Running Billing Cycle # 2
NOTICE:  Inserted 1 record(s) into billing table
NOTICE:  Inserted 1 record(s) into payment table
NOTICE:  Updated 1 record(s) in trips table and marked status as Processed
NOTICE:  Updated 1 record(s) in billing table and marked status as Processed
NOTICE:  Billing Cycle # 2 Completed Successfully
CALL
taxidb=>
```

```
WSParticipantRole:~/environment $
```

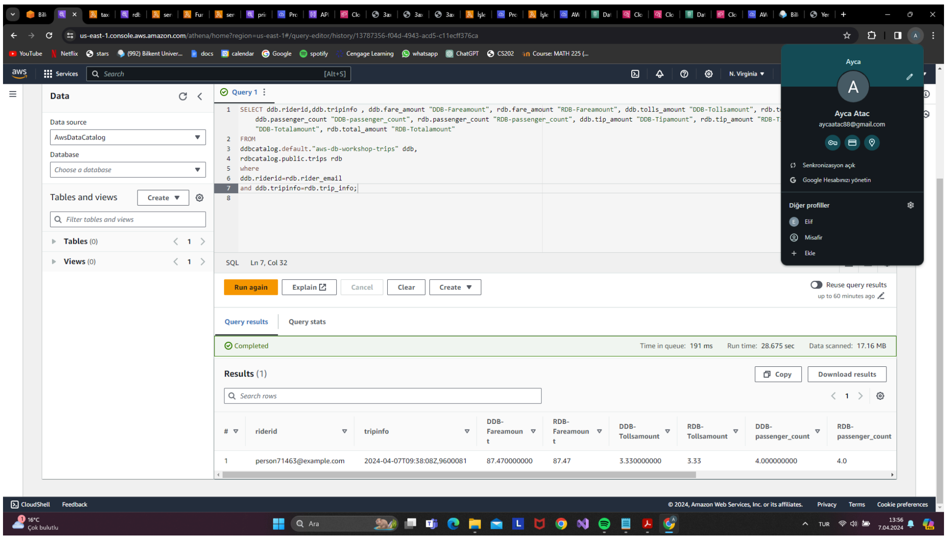
3. Lab3 - Query multiple data sources using Amazon Athena federated query

In the third part, I made use of the federated query service of Amazon Athena in order to execute queries for multiple different types of databases, in this case, DynamoDB and Aurora, also using S3 buckets. I created connectors that use Lambda functions which Athena will use to run queries. I then tested my implementations by entering queries in the Athena query editor and validated my results with the last query provided to finish the assignment.



Functions (6)

Function name	Description	Package type	Runtime	Last modified
taxiddb	Enables Amazon Athena to communicate with DynamoDB, making your tables accessible via SQL	Zip	Java 11	15 minutes ago
driver-complete-trip	Takes rider_id and trip_info as input and completes a trip by the driver by updating ws-db-workshop-trips table in DynamoDB	Zip	Python 3.8	1 hour ago
aws-db-workshop-ddb-stream-processor	Process completed taxi trip information from Amazon DynamoDB Streams and publishes the information to the trips table in Amazon Aurora database	Zip	Python 3.8	2 hours ago
taxirdb	Enables Amazon Athena to communicate with PostgreSQL using JDBC	Zip	Java 11	40 seconds ago
driver-accept-trip	Takes rider_id and trip_info as input and accepts a trip by the driver by updating ws-db-workshop-trips table in DynamoDB	Zip	Python 3.8	1 hour ago
rider-book-trip	Takes rider_id and rider_mobile as input and books a trip for the rider by updating ws-db-workshop-trips table in DynamoDB	Zip	Python 3.8	1 hour ago



```
SELECT ddb.riderid,ddb.tripinfo , ddb.fare_amount "DDB-Fareamount", rdb.fare_amount "RDB-Fareamount", ddb.tolls_amount "DDB-Tollsamount", rdb.t
    ddb.passenger_count "DDB-passenger_count", rdb.passenger_count "RDB-passenger_count", ddb.tip_amount "DDB-Tipamount", rdb.tip_amount "RDB-T
    "DDB-Totalamount", rdb.total_amount "RDB-Totalamount"
FROM
ddbcatalog.default."aws-db-workshop-trips" ddb,
rdbcatalog.public.trips rdb
where
ddb.riderid=rdb.rider_email
and ddb.tripinfo=rdb.trip_info;
```

SQL Ln 7, Col 32

Completed Time in queue: 191 ms Run time: 28.675 sec Data scanned: 17.16 MB

Results (1)

#	riderid	tripinfo	DDB-Fareamount	RDB-Fareamount	DDB-Tollsamount	RDB-Tollsamount	DDB-passenger_count	RDB-passenger_count
1	person71463@example.com	2024-04-07T09:38:08Z,9600081	87.470000000	87.47	3.330000000	3.33	4.000000000	4.0