Intermediate SQL

by Craig Sakuma

Introductions

Craig Sakuma

- Founder of QuantSprout
- General Assembly Instructor for Data Science
- MBA from Wharton
- B.Eng from Northwestern University

Fun Fact

Developed a novelty BBQ product that was featured in USA Today

Class Introductions

- Name
- What's your job?
- How do you plan to apply skills from today's workshop?
- Fun Fact

Objectives for Class

- Use SQL Workbench software to connect to database
- Import and export data
- Apply advanced querying techniques
- Basic database administration commands
- Build your own databases in the cloud

Course Structure

- Lectures on topics
  - Interaction is good
  - Feel free to ask questions
  - If there's not enough time to cover questions, we'll put it in a parking lot for after class
- Hands on exercises
  - Pair programming
  - Mix up partners

Set Up for Advanced SQL

- Download and install SQLWorkbench/J
  www.sql-workbench.net/downloads.html
- SQLWorkbench/J might require you to install or update your Java software
  java.com/en/download/
- Download the PostgreSQL JDBC driver (note: you won't be able to open this file)
  jdbc.postgresql.org/download.html

SQL Workbench

- Open Source software for communicating SQL commands to databases
- Compatible with a variety of SQL versions
- Connects to individual databases using credentials and server information
- Read and Save SQL queries as text files
- Easy to Import / Export data to and from CSV files

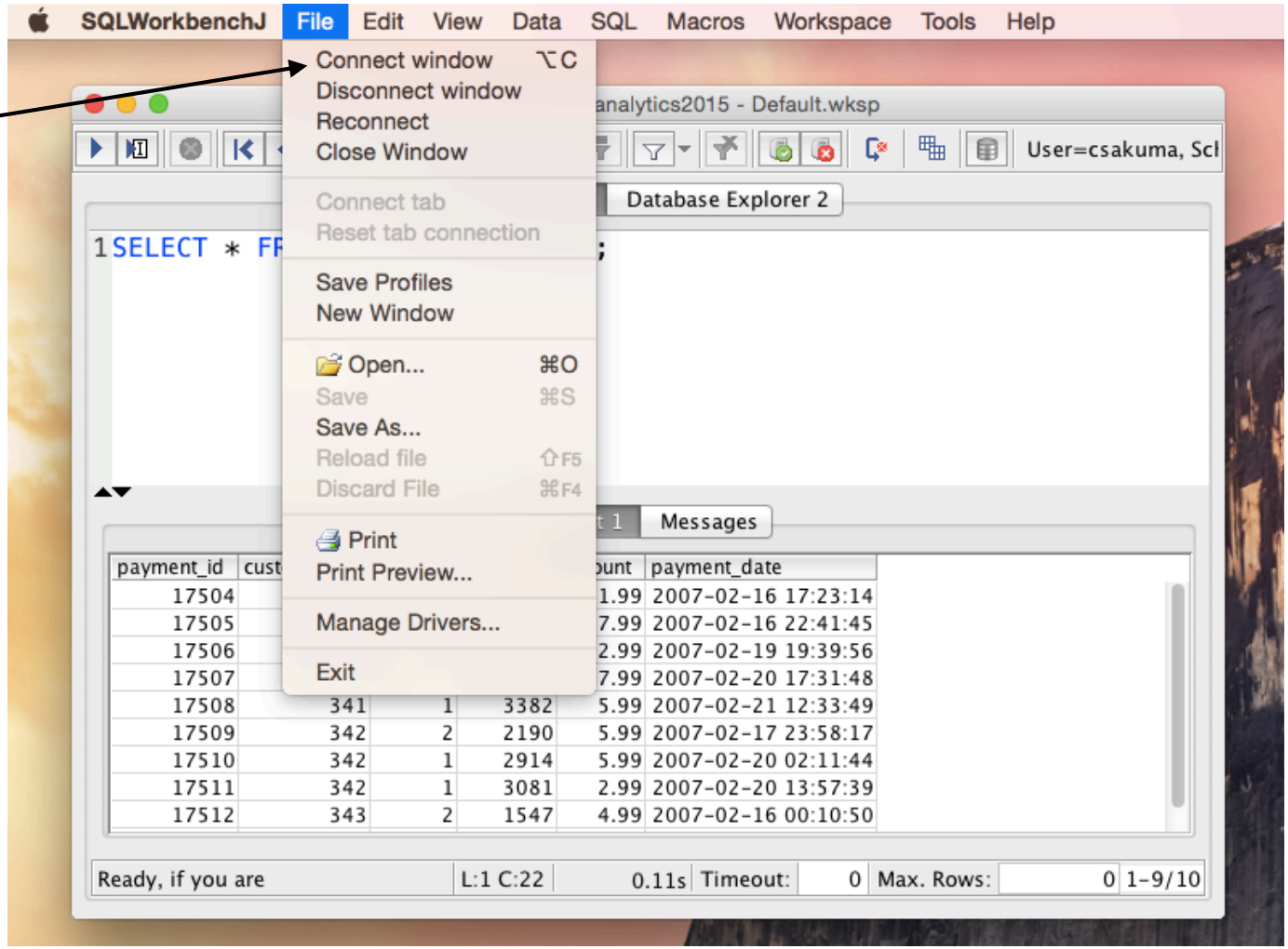
Download Driver

Driver files are used to communicate between SQL Workbench and each specific version of SQL (e.g., Postgres, MySQL, Oracle, etc…)

- Create folder called drivers in your home directory
- Download PostgreSQL driver
  - https://jdbc.postgresql.org/download.html
- Move file into your drivers folder

Open Connect Window

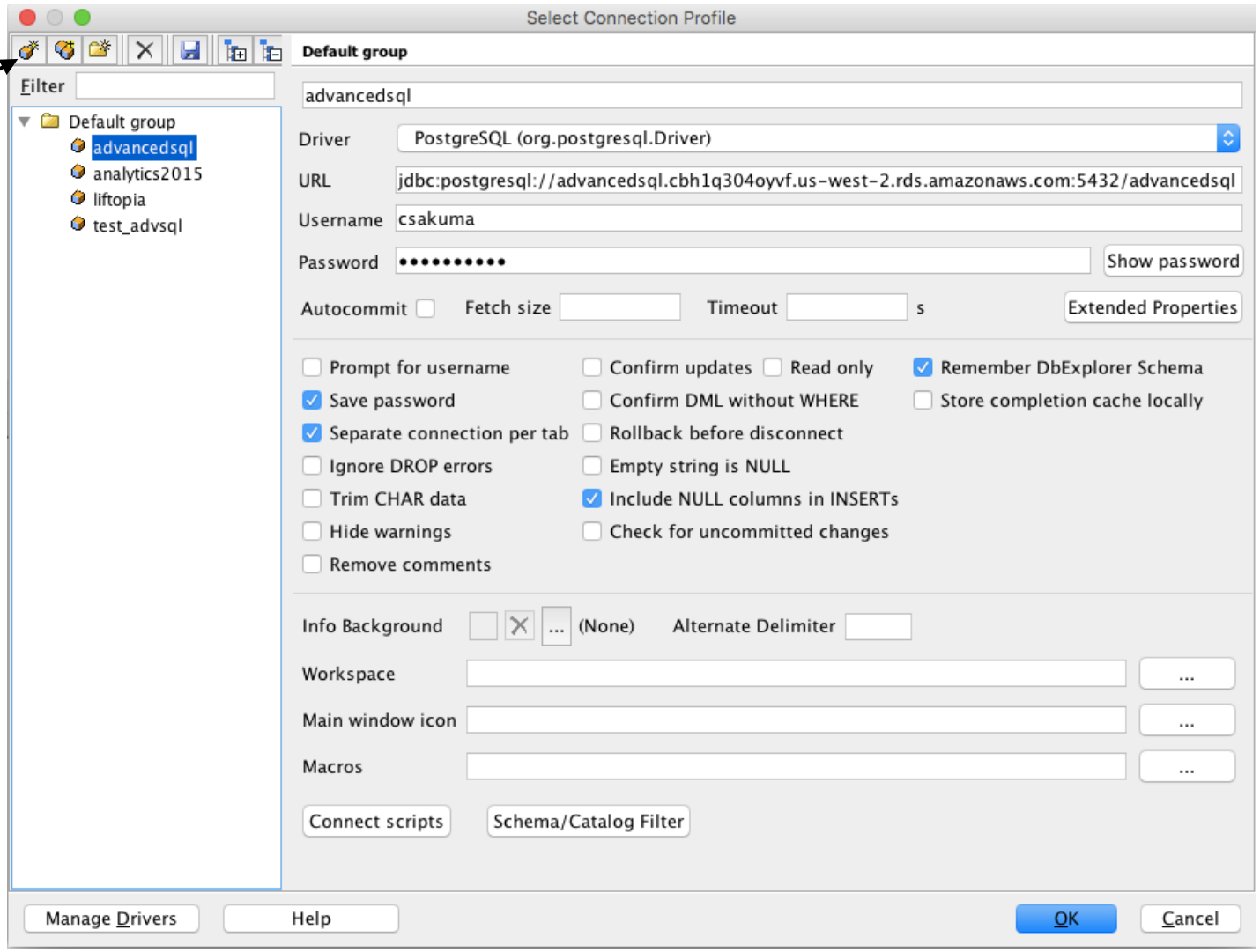
Select from File Menu



QuantSprout

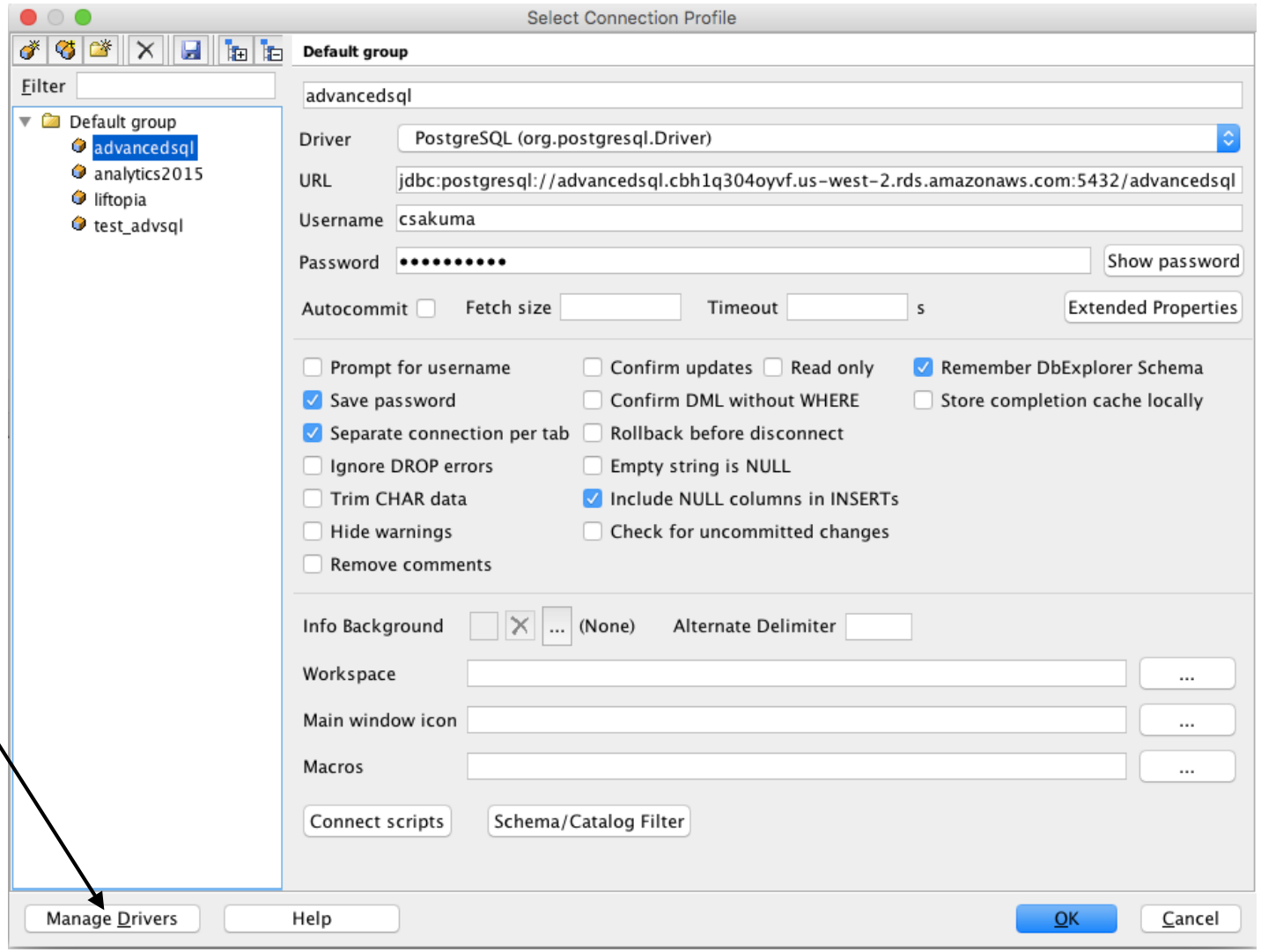
Create New Profile

Click to Create New Connection Profile



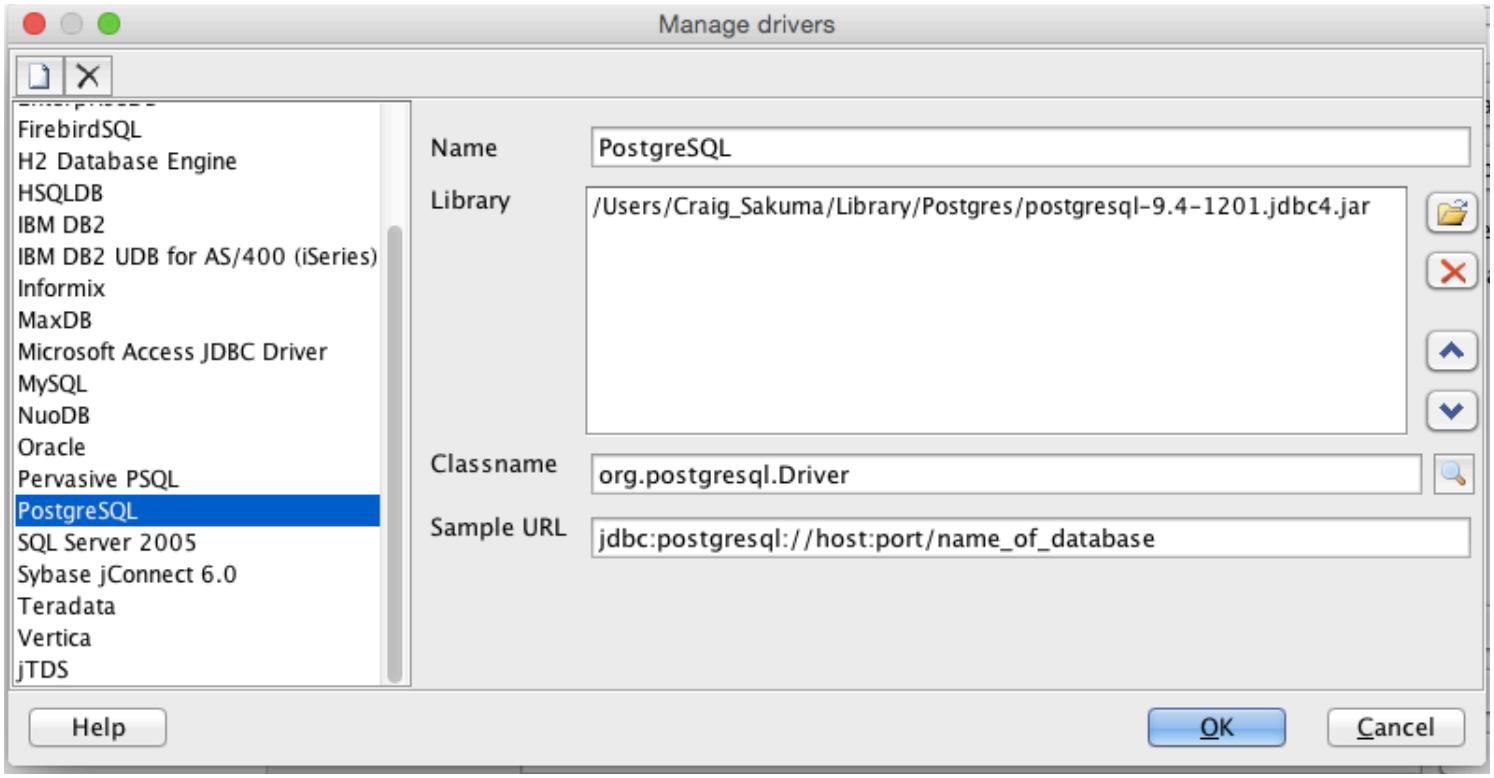
Update Driver

Click Manage Drivers

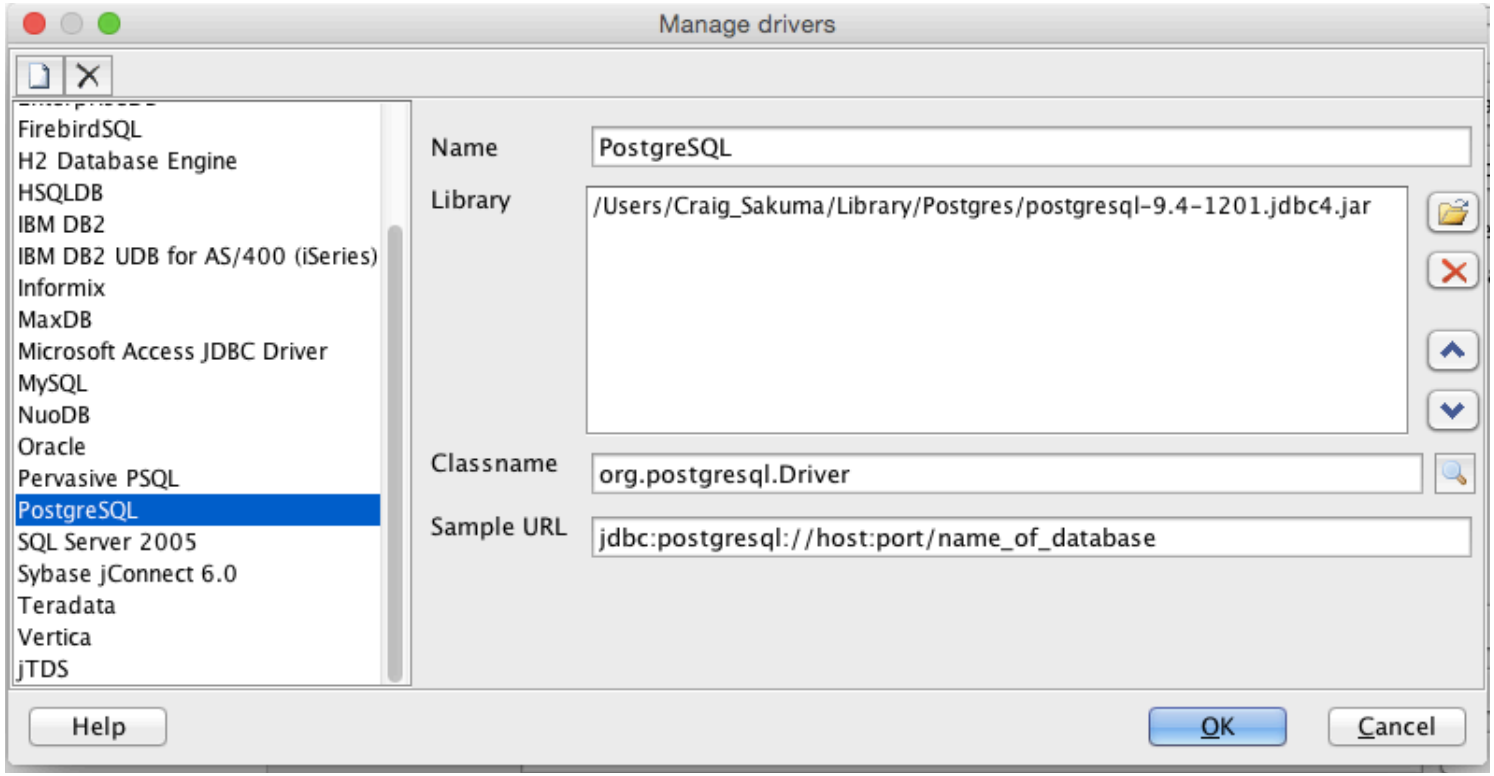


Update Driver

Select PostgreSQL



Update Driver



Click on folder icon and select Postgres Driver from your drivers folder

QuantSprout

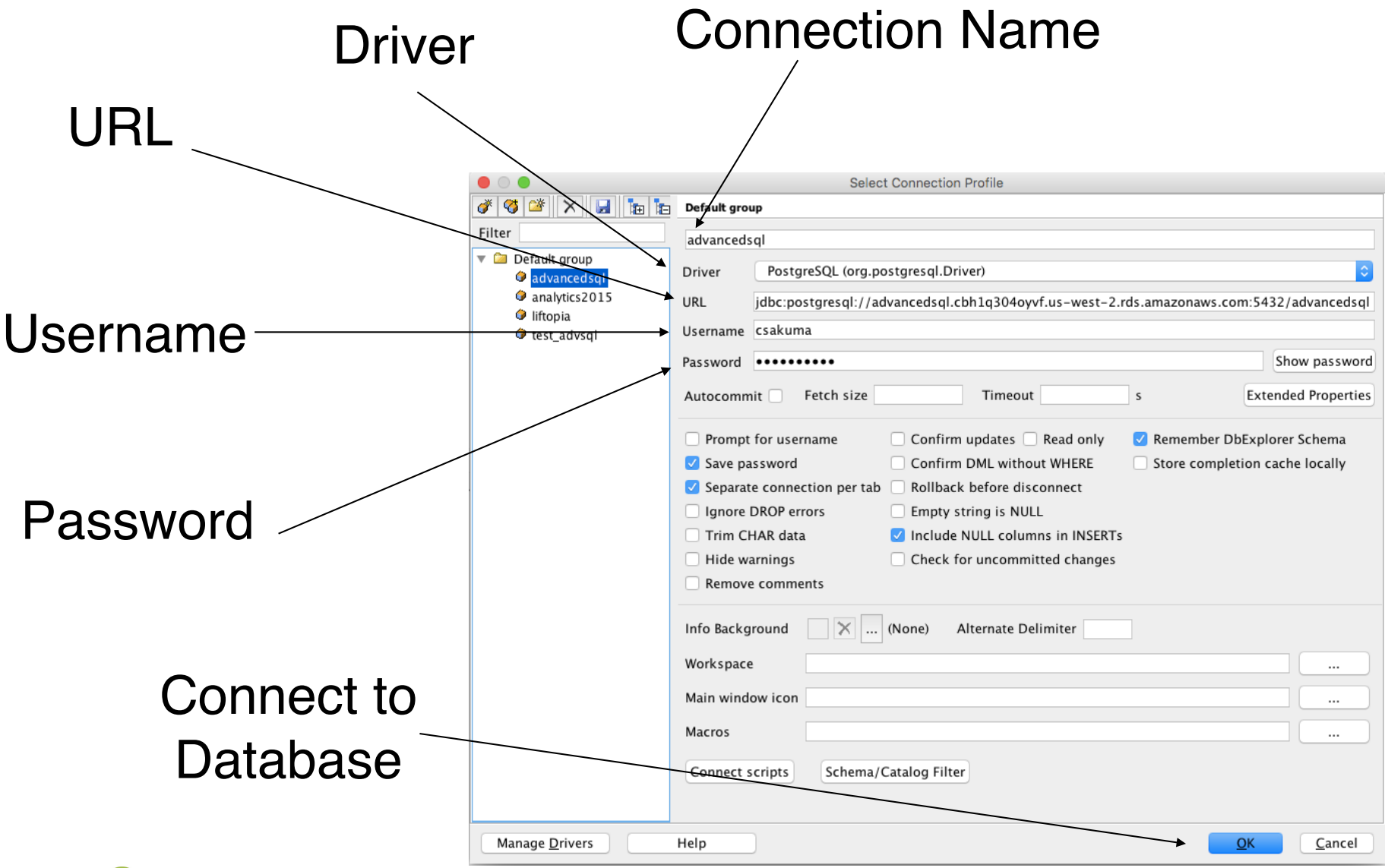
Connect to Database

- Set Up Connection (requires credentials)
  - Host
  - Port
  - Database Name
  - Username
  - Password
- URL for SQLWorkbench uses following format:
  - jdbc:postgresql://<Host>:<Port>/<DB Name>

Credentials for Database

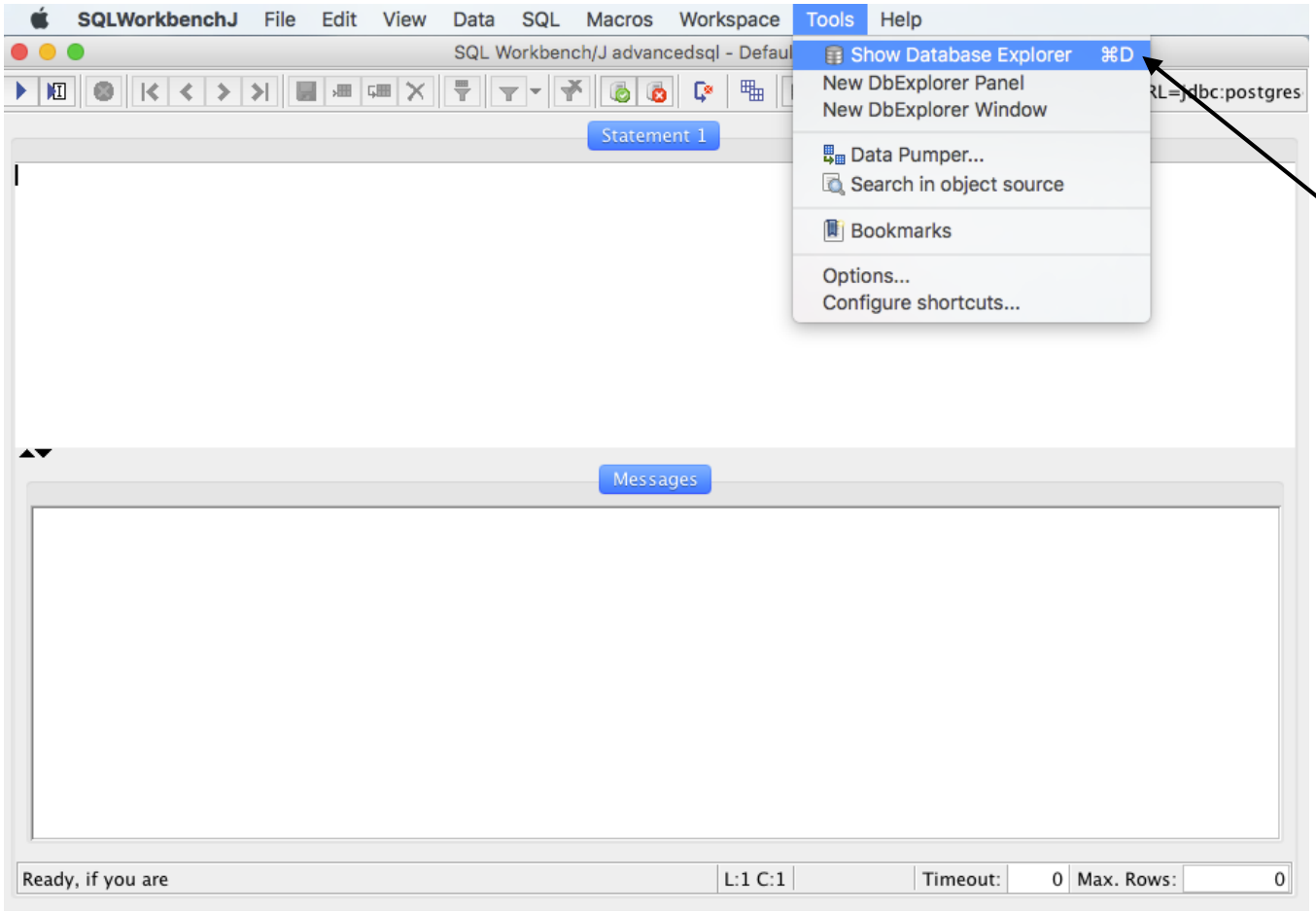
- URL:
  jdbc:postgresql://quantsprout.cbh1q304oyvf.us-west-2.rds.amazonaws.com:5432/QS
- User Name: your registration name
  (lower case with spaces replaced with underscore)
- Password: quantsprout

Add Credentials



QuantSprout

View Database Explorer



Select Database Explorer From Tools

Explore Database

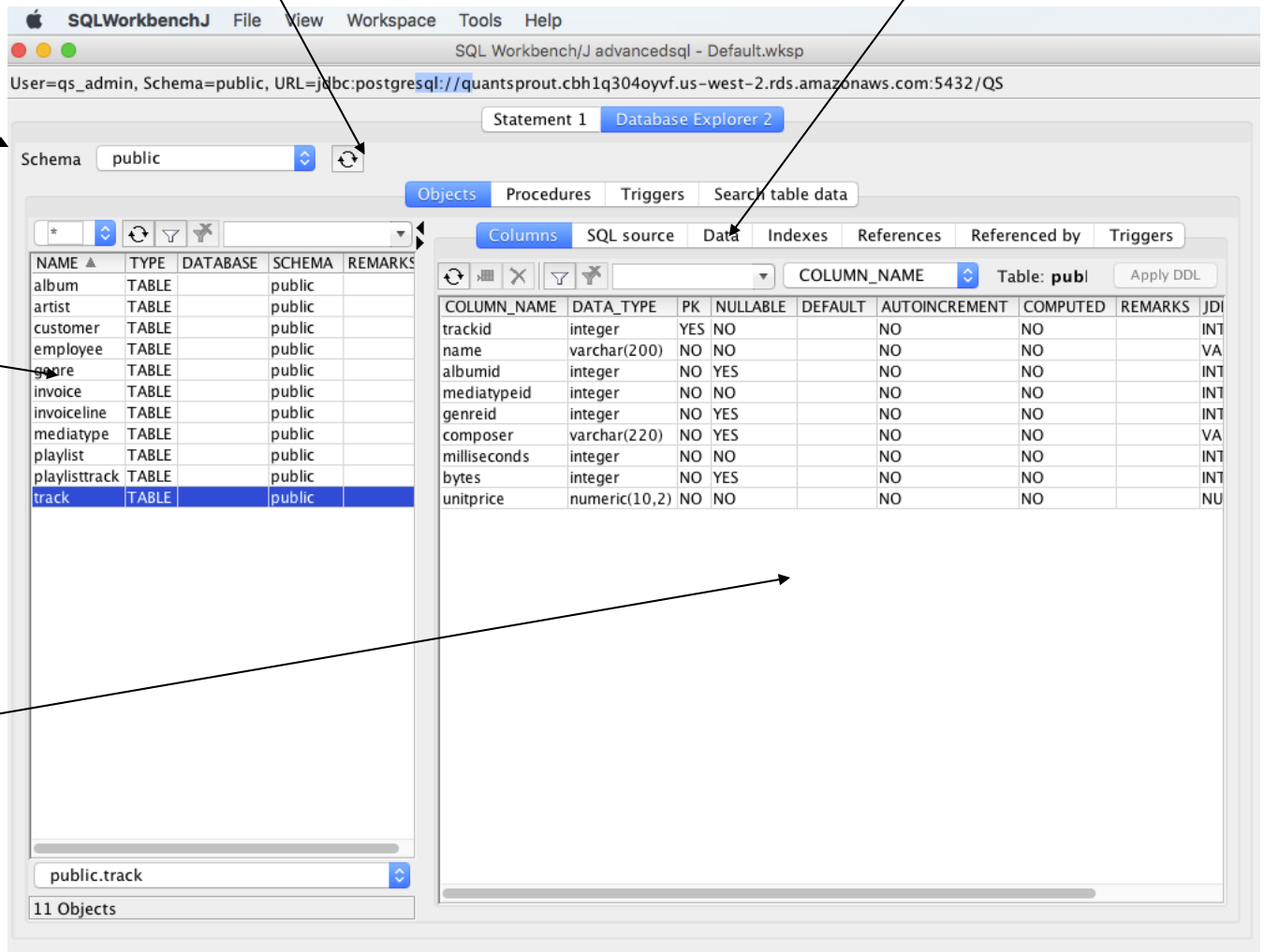
Select Schema

Refresh View

View Data Sample

Table List

Details for Selected Table

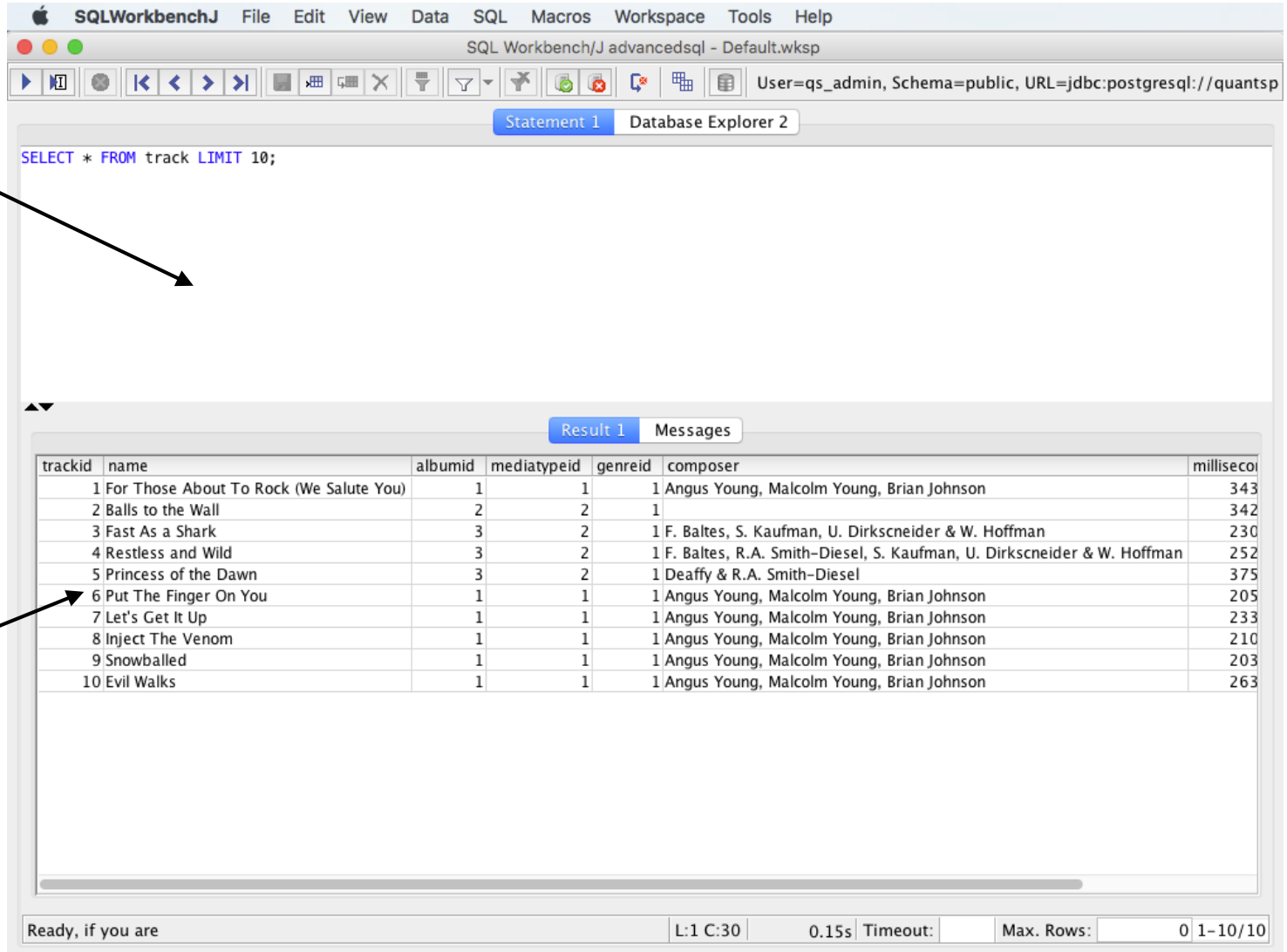


QuantSprout

Write Query

Text Editor for SQL code

Window for Results



QuantSprout

SQL – Comment

Comment out code that you don't want executed

-- <single line of text>

/* <multiple lines of text>

<multiple lines of text> */

Shortcut keys:

SQL workbench = **command + c + shift**

QuantSprout

Review Query Commands

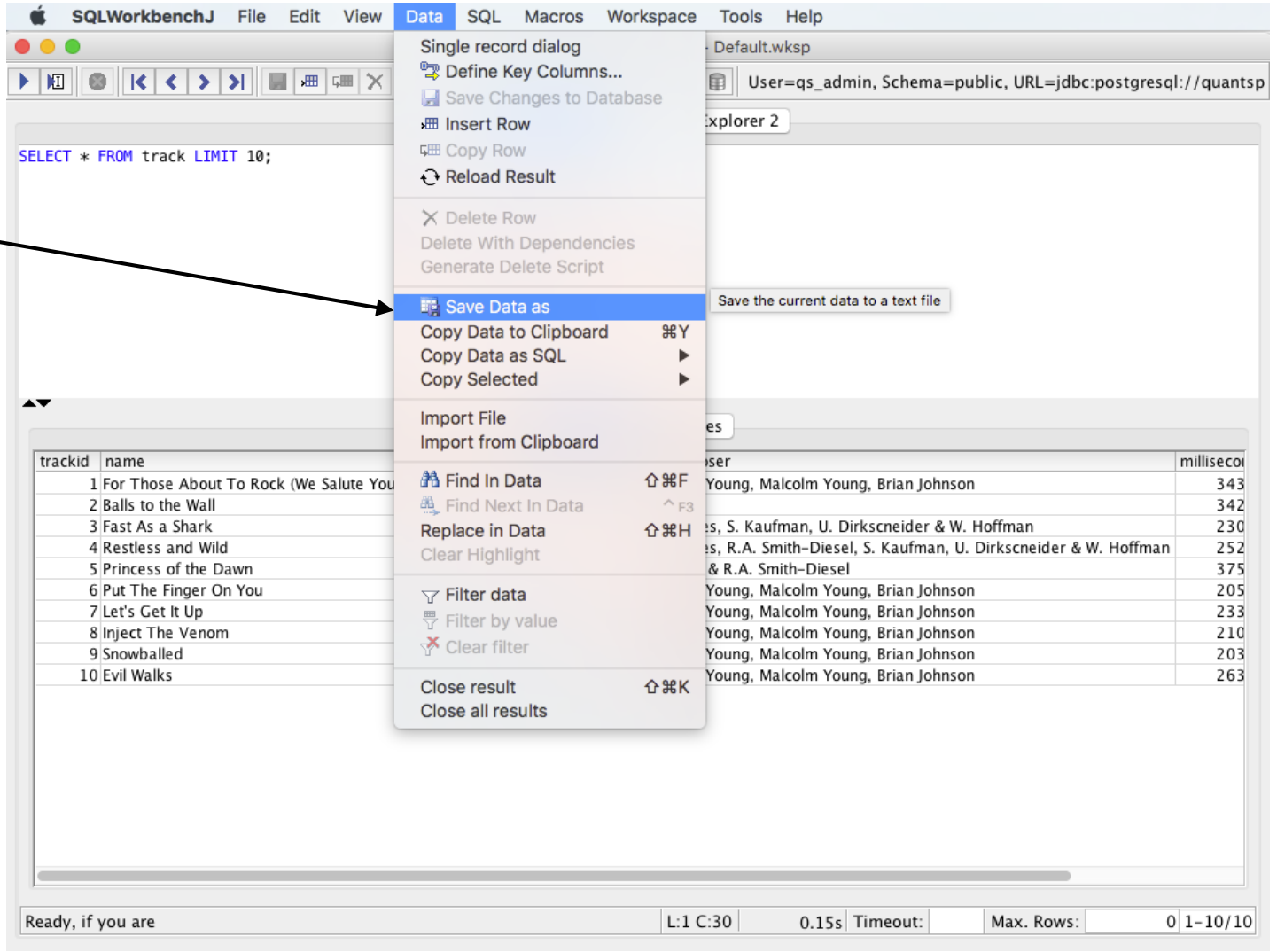
- **SELECT**: choose the fields for query
- **FROM**: pick table(s) for data source
- **WHERE**: filter data based upon conditions
- **GROUP BY**: segment data into groups
- **ORDER BY**: sort results
- **LIMIT**: limit the number of records returned
- **JOIN Types**:
  - INNER vs. OUTER
  - LEFT vs. RIGHT

Review Exercise

- How many days of content are there in the library?
- What are the longest songs (excluding video)?
- What is the average length of a song grouped by genre (convert time to minutes)?

Export Data

Select Save Data



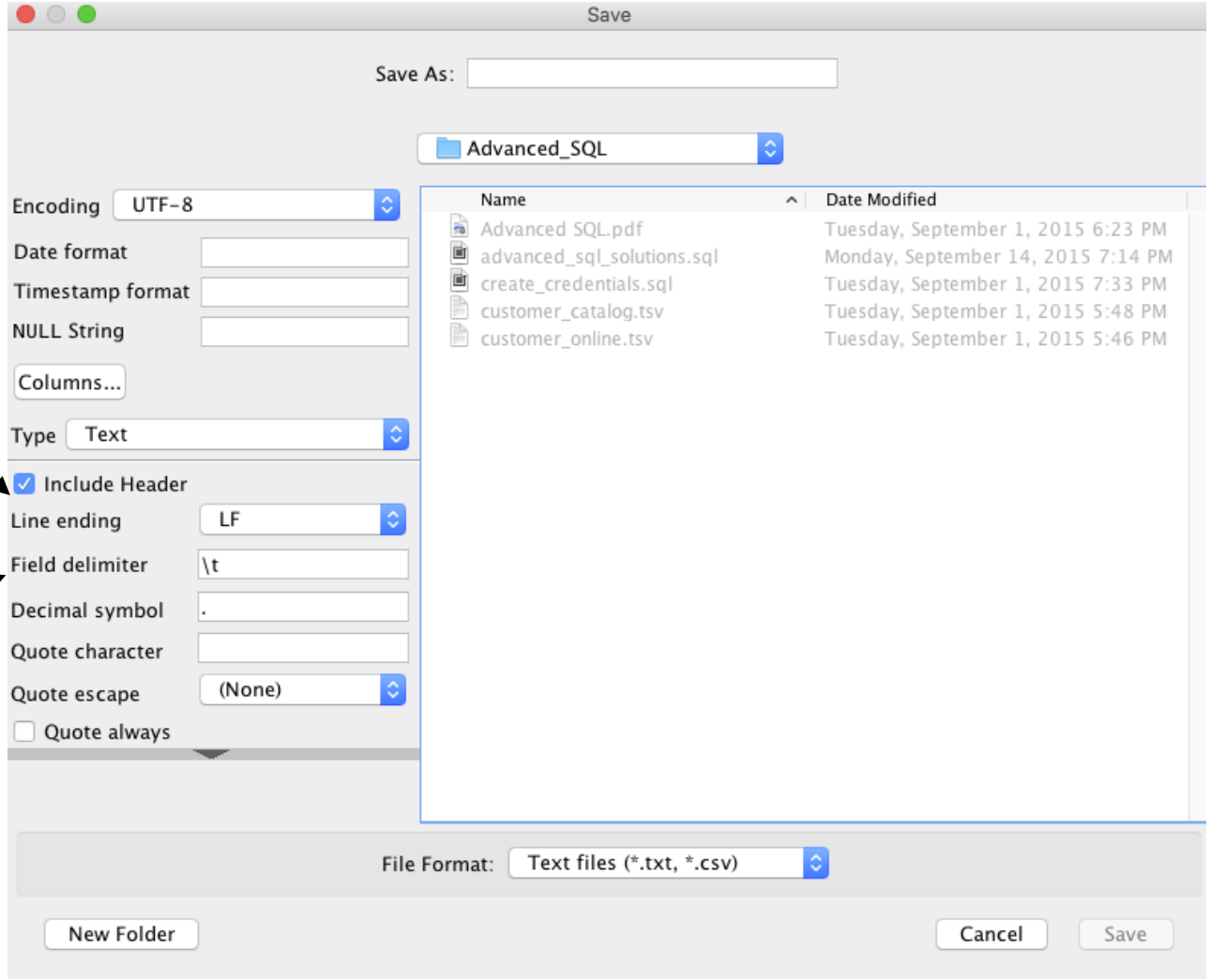
QuantSprout

Export Data

Option for Headers

Pick Delimitter

- tab: \t
- pipe: |
- comma: ,



Aliases

Rename fields in your queries:

- SELECT <field> AS <alias>
- SELECT milliseconds/1000 AS seconds

Reference tables as abbreviations:

- FROM <table> <alias>
- FROM track t
  JOIN genre g
  ON t.genreid = g.genreid

Aliases

Reference fields by position:

- SELECT <field1>, <field2>
  FROM <table>
  ORDER BY 1, 2
- SELECT name, milliseconds
  FROM track
  ORDER BY 2 DESC

Extract

- Syntax:
  SELECT EXTRACT (<date_component> FROM <field>)

- Example:
  SELECT EXTRACT(month FROM invoicedate)
  FROM invoice;

- Date/Time Components

Day	Hour	DOW
Week	Minute	DOY
Month	Second	Quarter
Year		Timezone

Having

- Syntax:
  SELECT <field1>, <agg function>(<field2>)
  FROM <table>
  GROUP BY <field1>
  HAVING <agg function>(<field2>) <operator> <value>;

- Example:
  SELECT genreid, COUNT(trackid)
  FROM track
  GROUP BY genreid
  HAVING COUNT(*) > 50;

Exercise #1

- Which customers have spent more than $40 (Use Group By and Having for the **customer** and **invoice** tables)?
- What are the total sales by month (Use Extract and Group By and the **invoice** table)?
- Create a roster of employees with their bosses (Join the **employee** table to itself by using table aliases)

Case Statements

- Apply conditional logic

```
SELECT CASE
    WHEN <condition1> THEN <result1>
    WHEN <condition 2> THEN <result 2>
    ELSE <result 3>
END
```

- Example:

```
SELECT CASE
    WHEN age < 18 THEN ‘child’
    WHEN age >= 60 THEN ‘senior’
    ELSE ‘adult’
END AS age_market_segment
```

Union

- Merges data from two queries by stacking results on top of each other
- Must have same number of columns and corresponding data types
- Duplicate results are removed by default
- UNION ALL will include duplicates

- Syntax:
  SELECT * FROM <table1>
  UNION
  SELECT * FROM <table2>

Coalesce

Picks first non-null value

- Syntax:
  COALESCE ([field1], [field2], [field3])

- Example:
  COALESCE(online.firstname, catalog.firstname) as firstname

Exercise #2

- Using the **iowa liquor products** table, create an alcohol type label for whisky, vodka, tequila, rum, brandy, schnapps and any other liquor types (hint: use CASE STATEMENT and LIKE)
- Using the **catalog** and **online** tables, create a customer list that combines the names from the catalog and online tables using UNION without creating duplicates.
- FULL JOIN the **catalog** and **online** tables and inspect the results. Try adding the catalog sales and online sales totals together. Why do you get errors?

Window Functions

- Perform calculation across a set of table rows that are somehow related to the current row
- Applications:
  - Cumulative sales
  - Percentile rank
  - Group level results

Cumulative Sales

- Syntax:
  SELECT sum(<field1>) OVER
  (ORDER BY <field2>)
  FROM <table>;

- Example:
  SELECT invoicedate, SUM(total)
  OVER (ORDER BY invoicedate)
  FROM invoice;

Percentiles

- Syntax:
  SELECT NTILE(<# of groups>) OVER
  (ORDER BY <field>)
  FROM <table>;

- Example:
  SELECT name, milliseconds, NTILE(100)
  OVER (ORDER BY milliseconds DESC) AS
  percentile
  FROM track;

Group Level Results

- Syntax:
  SELECT AVG(<field1>) OVER
  (PARTITION BY <field2>)
  FROM <table>;

- Example:
  SELECT name, genreid, milliseconds,
  AVG(milliseconds)
  OVER (PARTITION BY genreid)
  FROM track;

Subqueries

- Syntax:
  WITH <subquery_name> AS (
      <code_for_subquery>)
  <code for parent query>;

- Example:
  WITH top_customers AS (
      SELECT customerid, SUM(total) as sales
      FROM invoice
      GROUP BY customerid)
  SELECT COUNT(*)
  FROM top_customers
  WHERE sales > 40;

Views

- Syntax:,
  CREATE VIEW <view_name> AS <query_code>;
- Example:
  CREATE VIEW top_customers AS
  SELECT customerid, SUM(total) as sales
  FROM invoice
  GROUP BY customerid
  ORDER BY sales DESC;

NOTE: You can't have duplicate field names in a view

Temporary Tables

- Syntax:
  CREATE TEMP TABLE <table_name> AS
  <query_code>;

- Example:
  CREATE TEMP TABLE total_sales AS
  SELECT customerid, SUM(total) as sales
  FROM invoice
  GROUP BY customerid
  ORDER BY sales DESC;

Exercise #3

- How many iowa liquor vendors have more than $1 million in 2014 sales (hint: use subquery to group sales by vendor)?
- Group sales by month with a subquery and then calculate cumulative sales by month for 2014 (using iowa sales table)
- Create a View that adds liquor type to the iowa product data. Don't forget to commit your changes.

Regular Expressions

- Programming language for pattern matching text
- Similar in functionality to LIKE but much more powerful

- Syntax:
  SELECT * FROM [table_name]
  WHERE [field1] ~ [pattern];

Regular Expressions

- Example (all artists with names ending with n):
  SELECT * FROM artist
  WHERE name ~ ‘n$’;

- Example (all artists that start with z):
  SELECT * FROM artist
  WHERE name ~ ‘^z’;

Amazon Web Services

- Cloud services that can be rented
  - Pay based on usage (e.g., hours of use, volume of data)
  - Simple to set up and shutdown
  - Services are very scalable
  - Excellent security options

- Wide range of products:
  - Virtual Servers in the Cloud – EC2 (Elastic Compute Cloud)
  - Cloud file storage – S3 (Simple Storage Service)
  - Databases as a service – RDS (Relational Database Service)
  - Data Warehouses – Redshift

AWS Relational Database Service

- Set Up
  - Provision server in the cloud (e.g., EC2)
  - Download and install software
- Database Administration
  - Allocate storage space for database
  - Backup of data
  - Mirror data across multiple locations
- Database Maintenance
  - Upgrade software when new releases and patches
  - Scale database as data grows

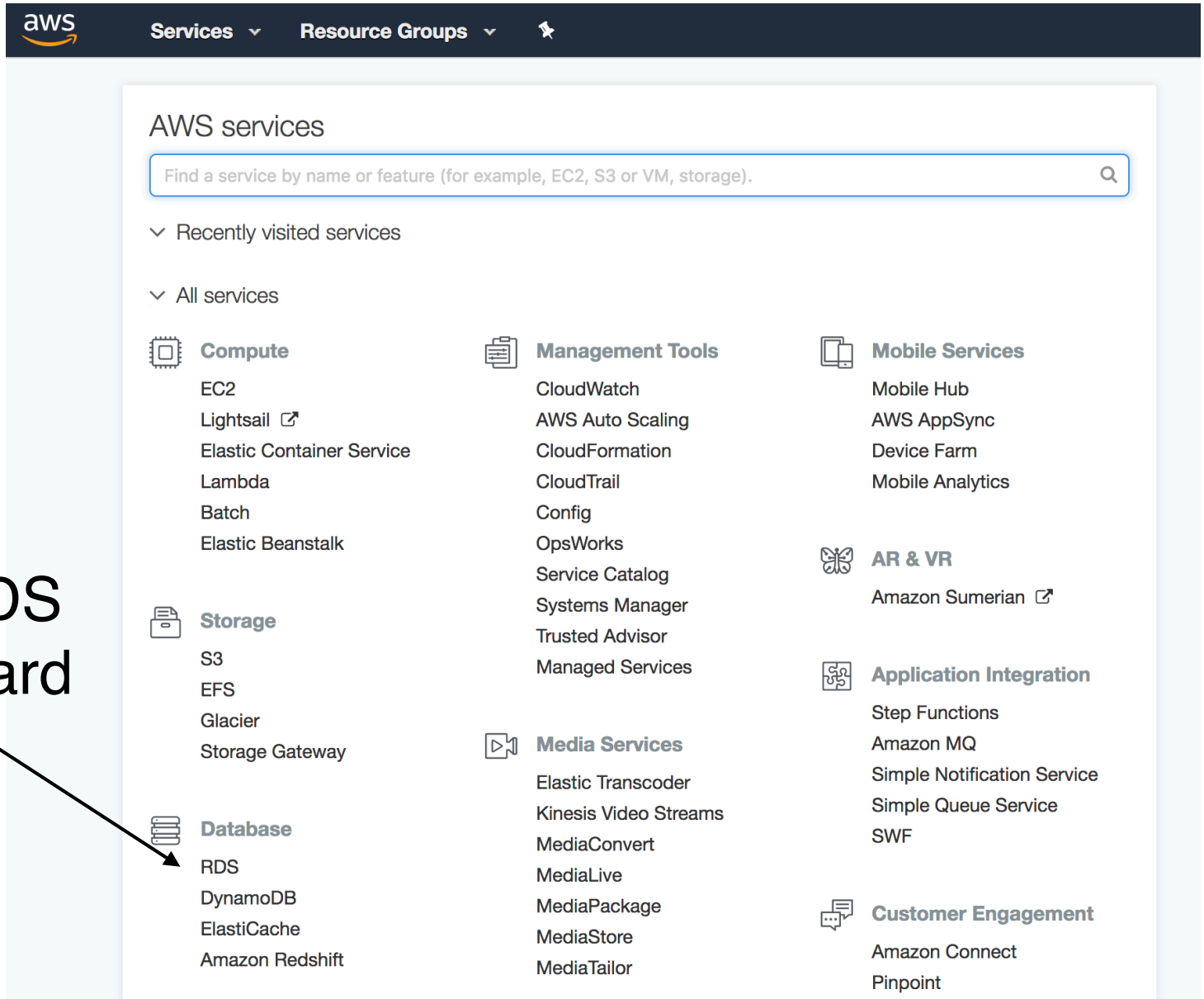
Outsource Database Operations to Amazon

Create AWS Account

Follow registration process at aws.amazon.com:

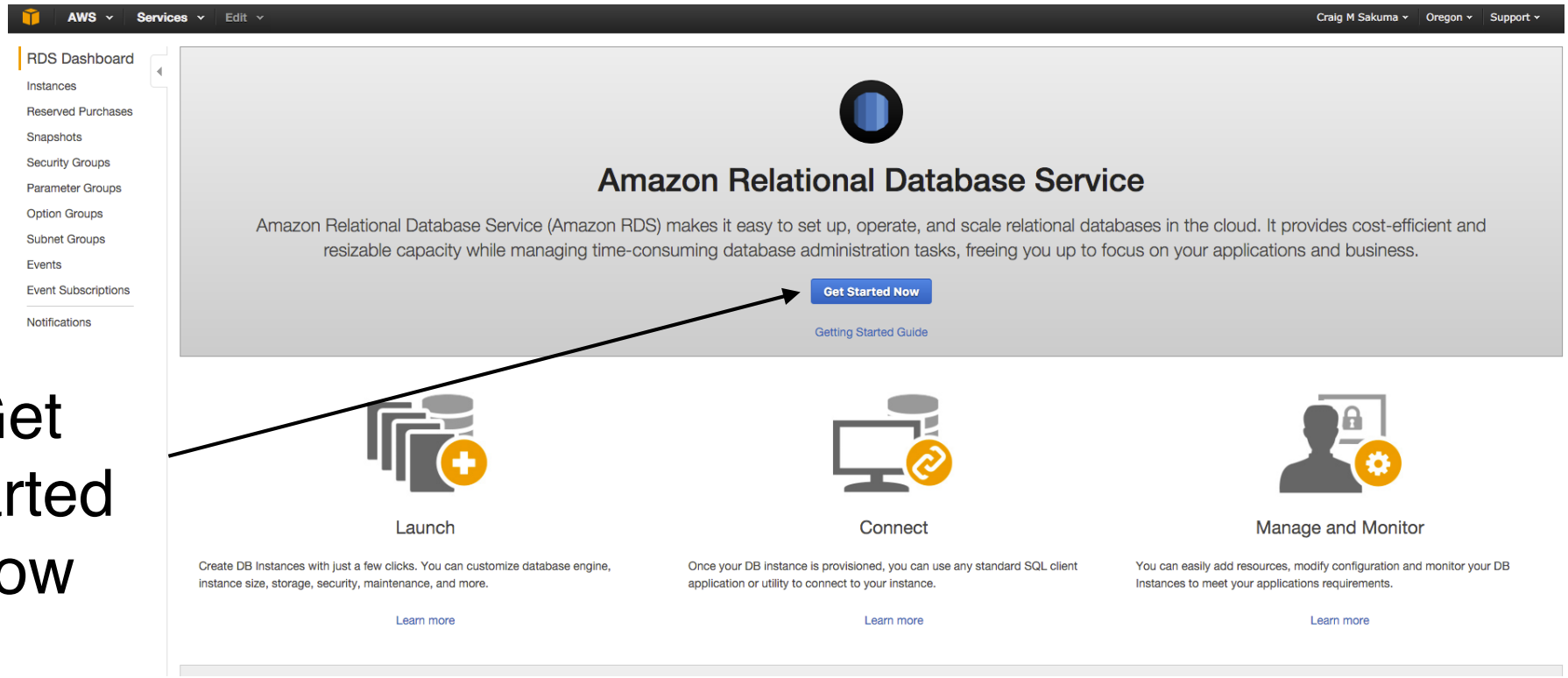
- Create User ID and Password
- Add contact info
- Add credit card info (we'll be using free service so you won't be charged)
- Add phone number and verify identity via call
- Select Basic(Free) support plan
- Launch Management Console

AWS Console



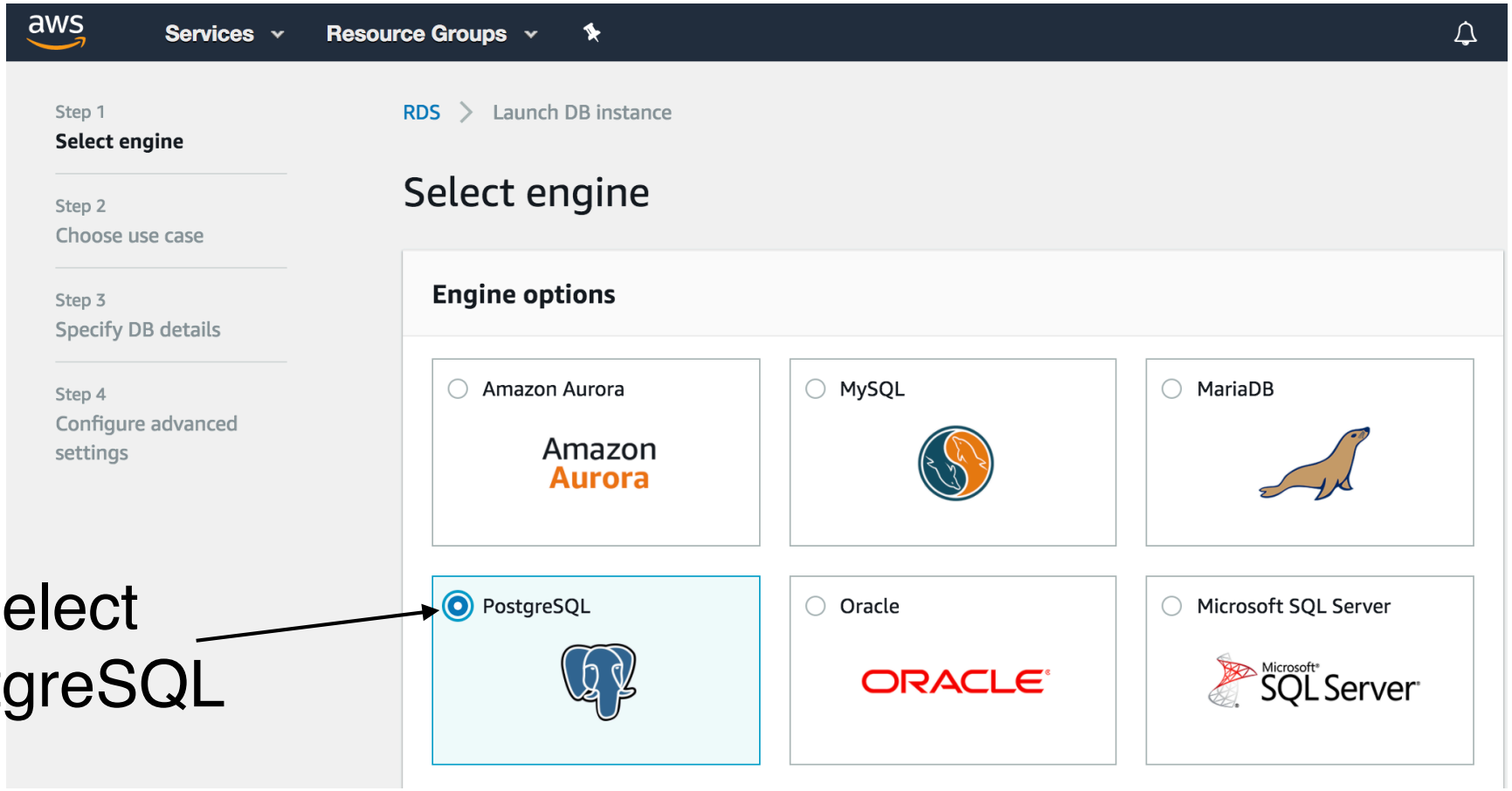
View RDS Dashboard

RDS Dashboard



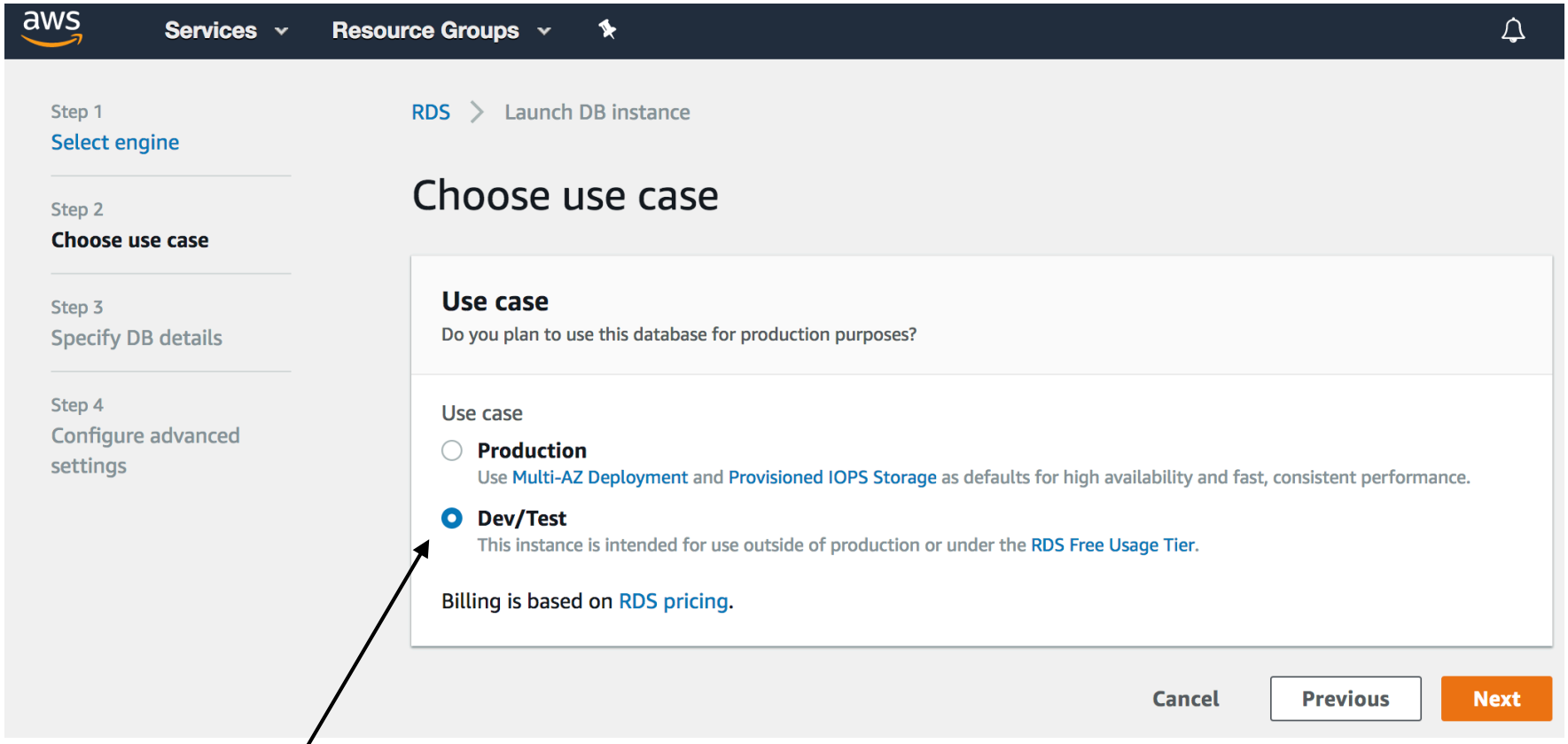
Get Started Now

Select Version of SQL



Select PostgreSQL

Multi-AZ Deployment

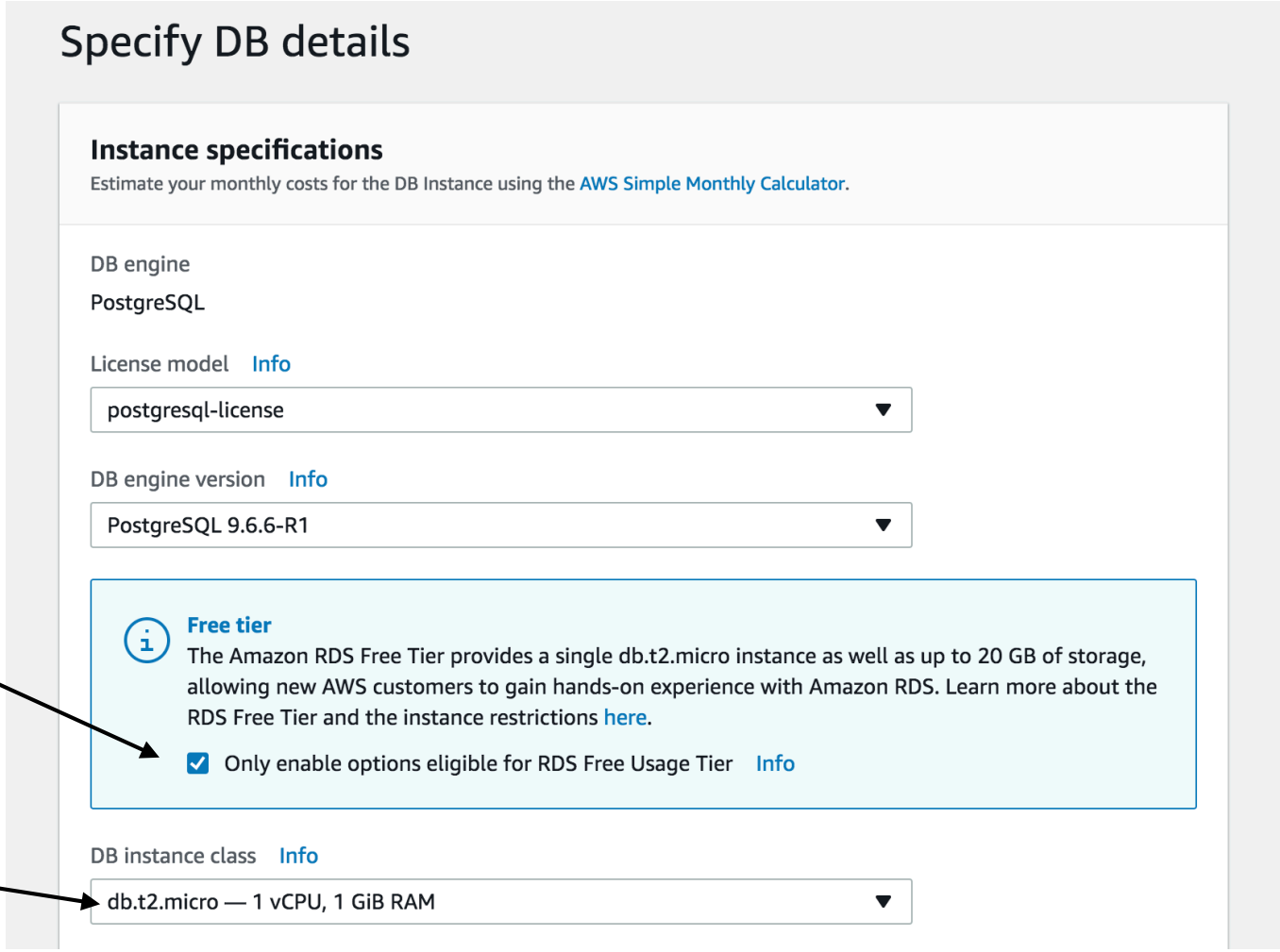


Choose Dev/Test

Specify DB Details

Check
Free tier

Select
db.t2.micro

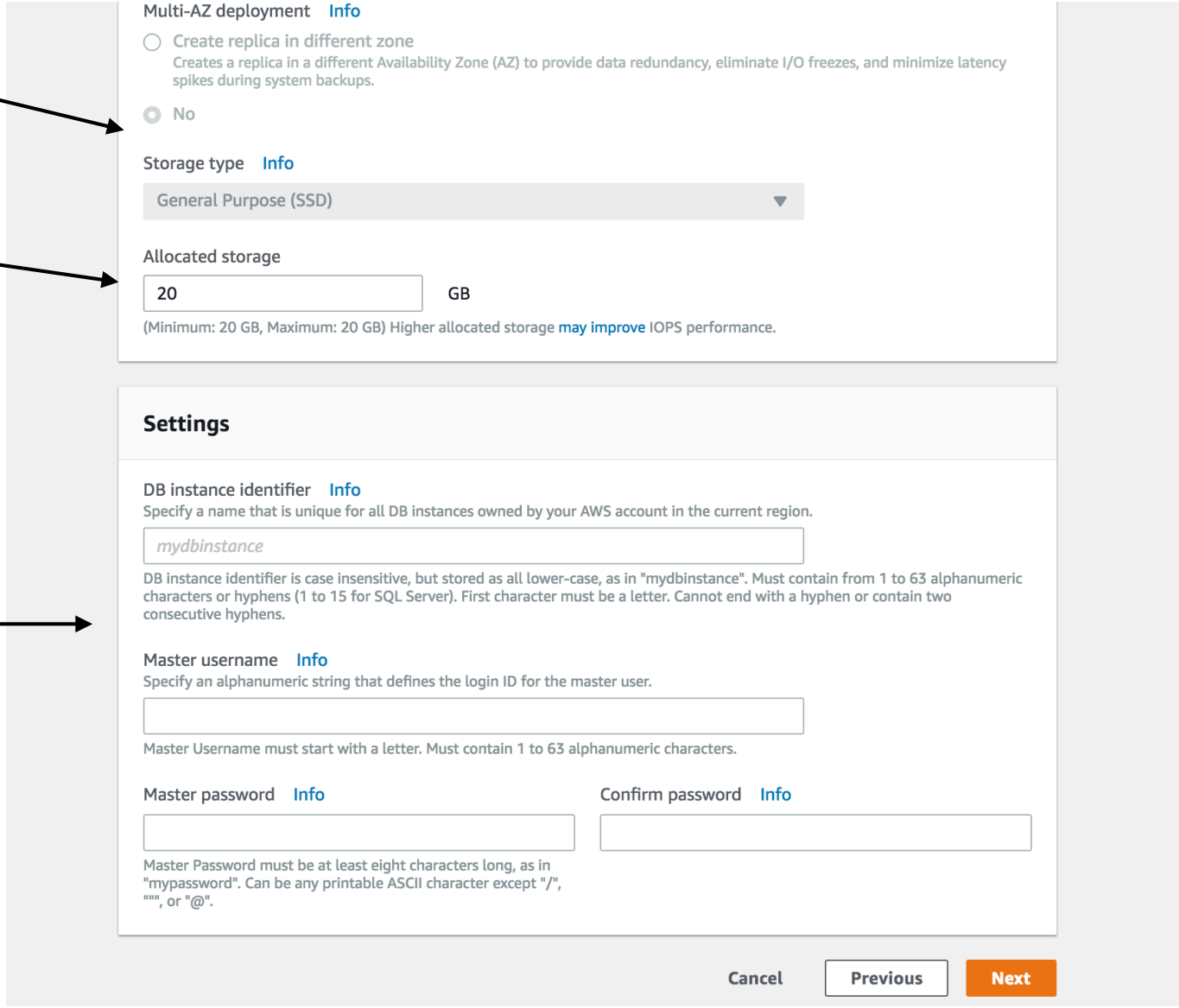


Specify DB Details

No Multi-AZ

20GB Storage

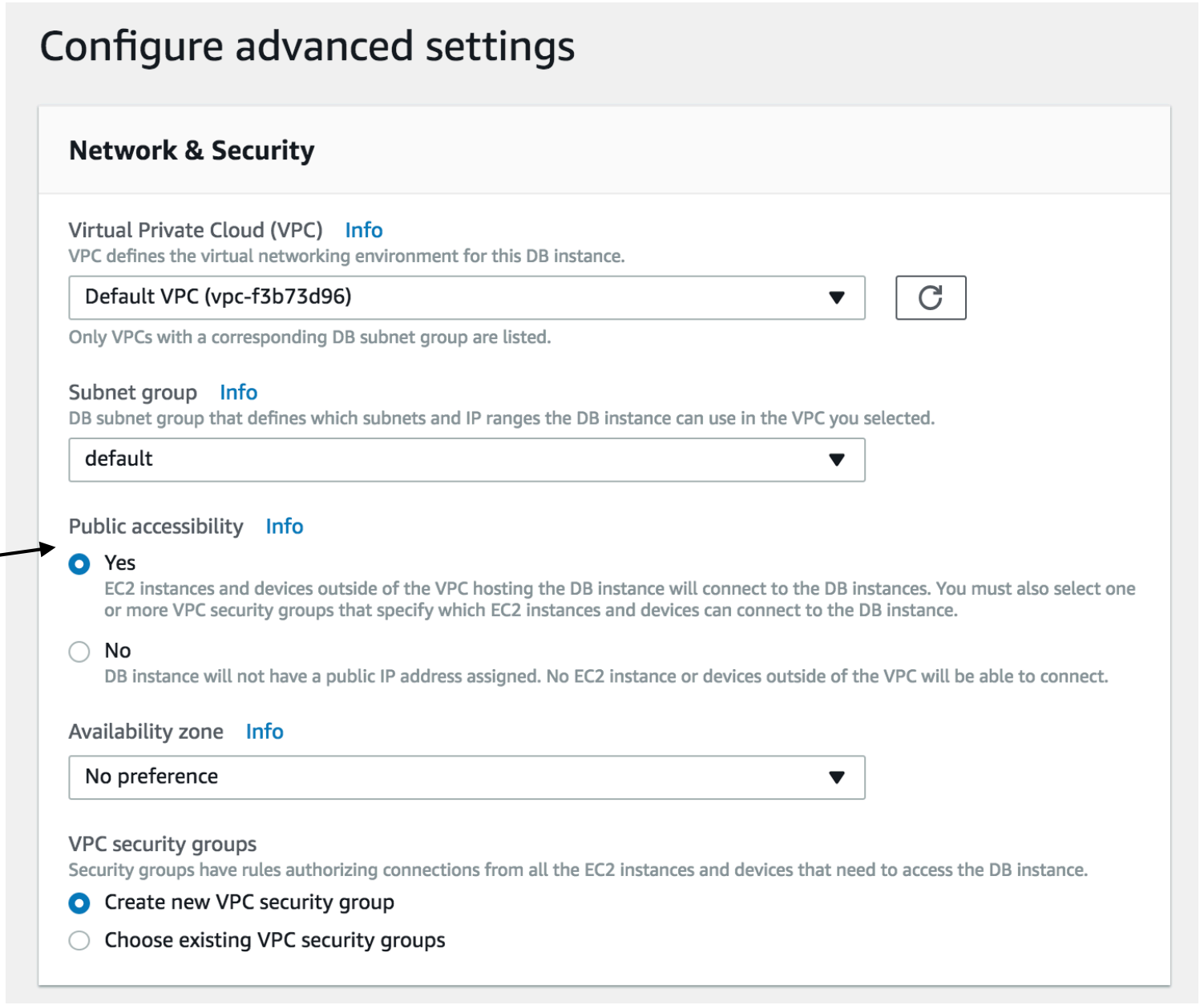
Add Identifier, Username and Password



QuantSprout

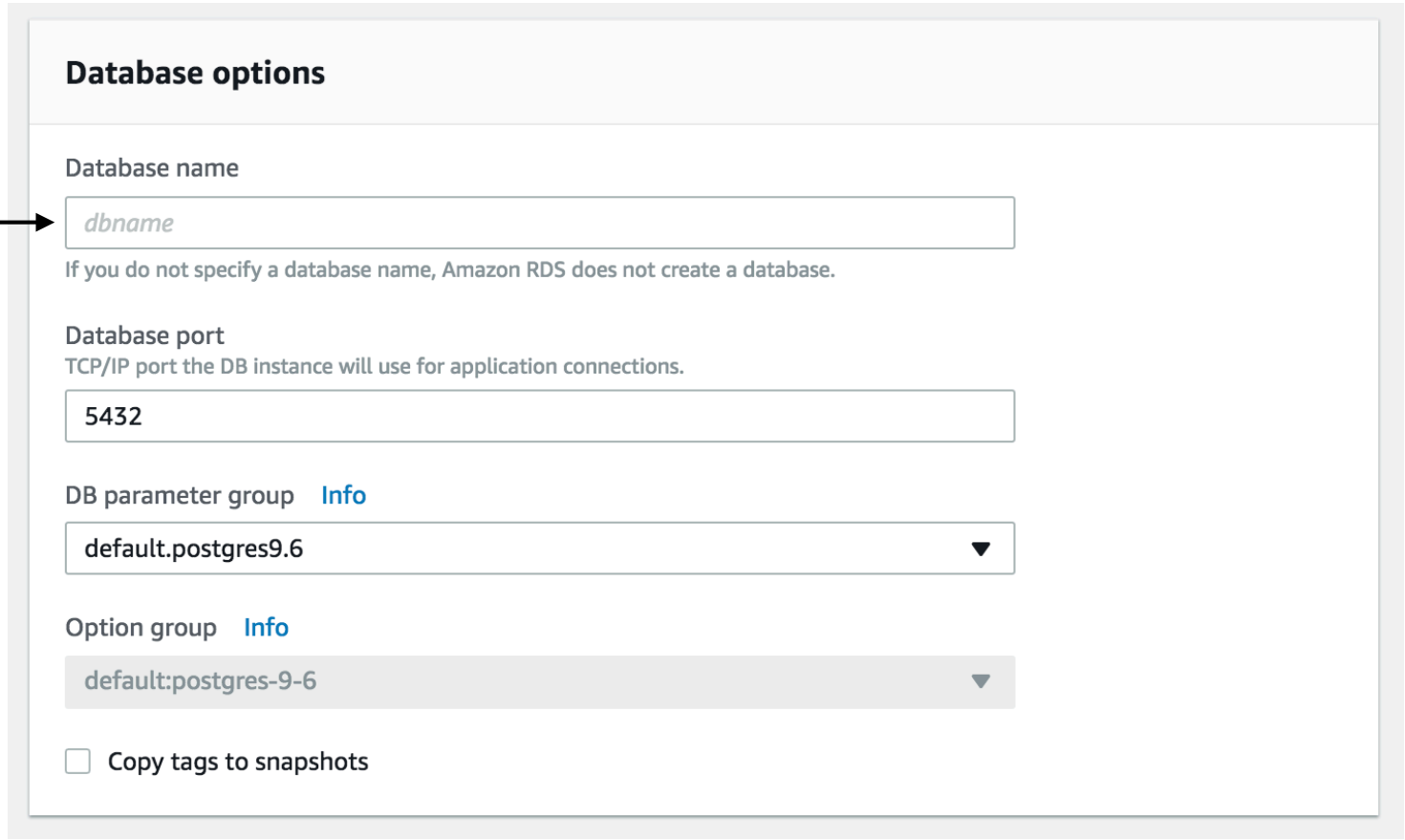
Advanced Settings

Check for Public accessibility

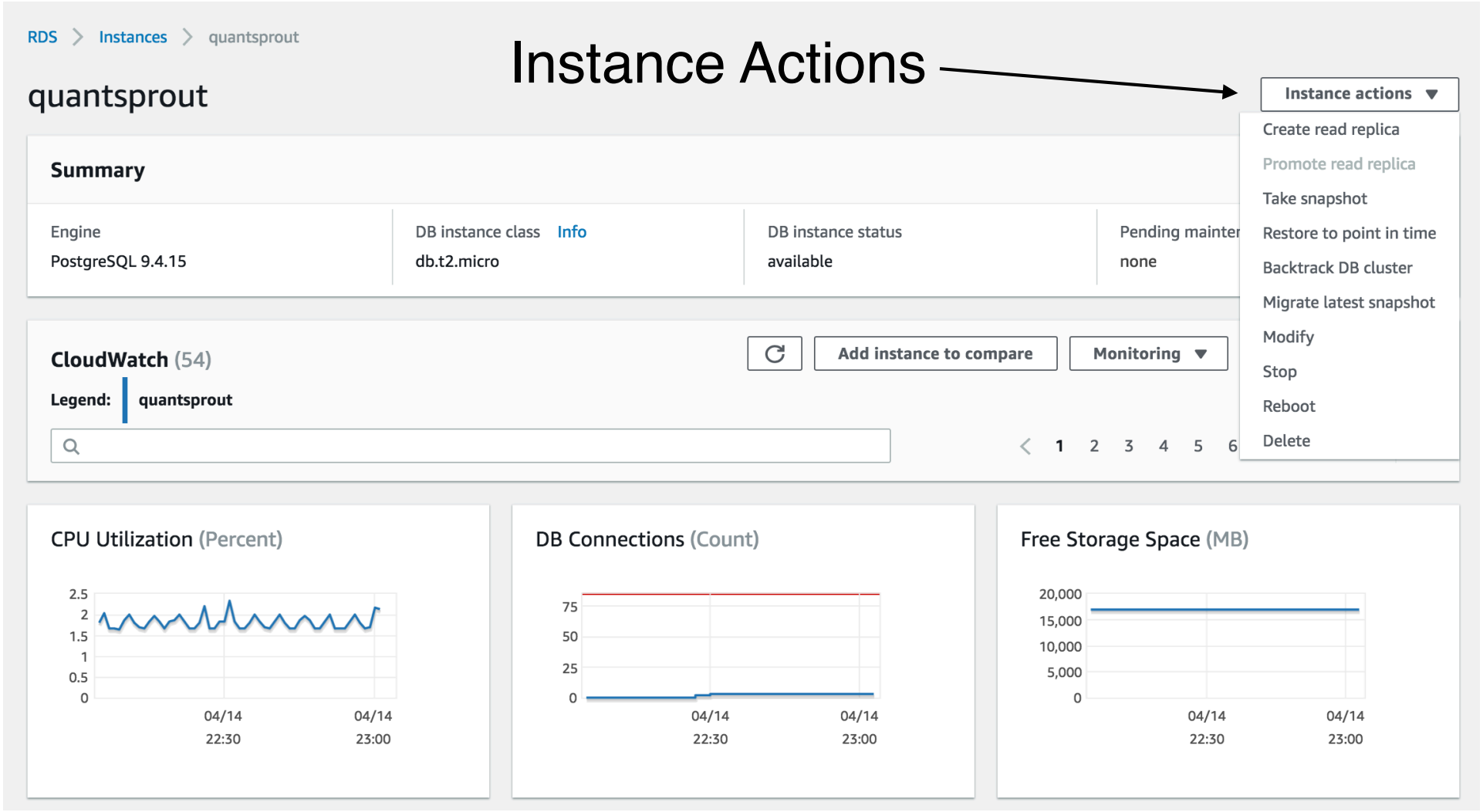


Advanced Settings

Add Database Name



View Database Details

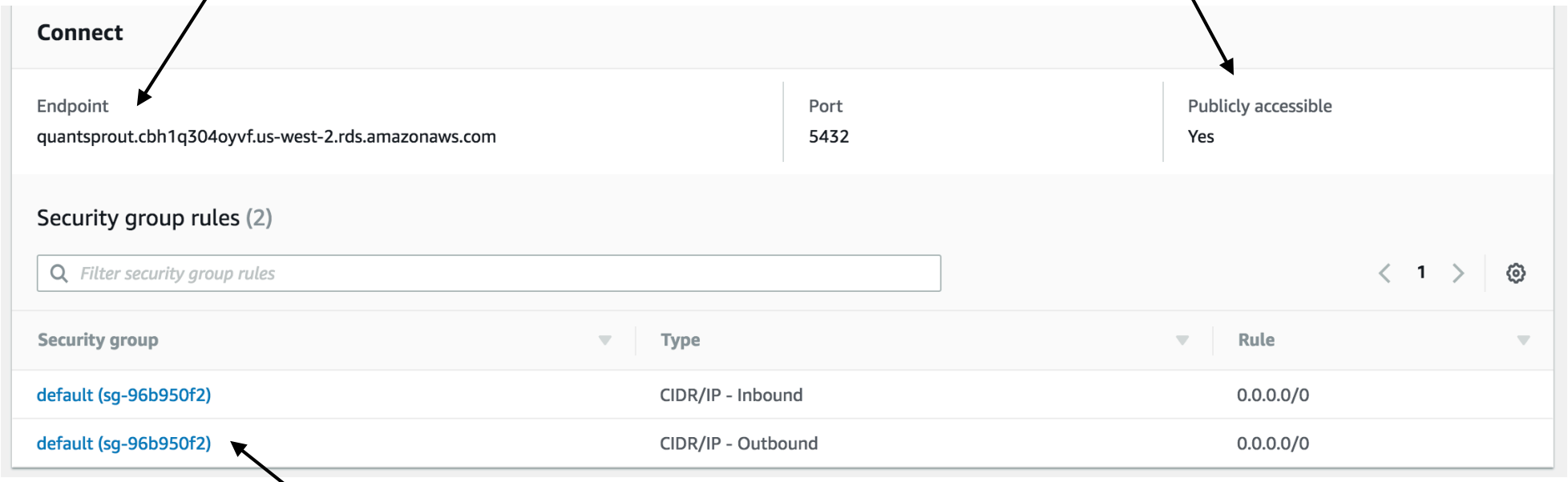


QuantSprout

View Database Details

Endpoint for Host

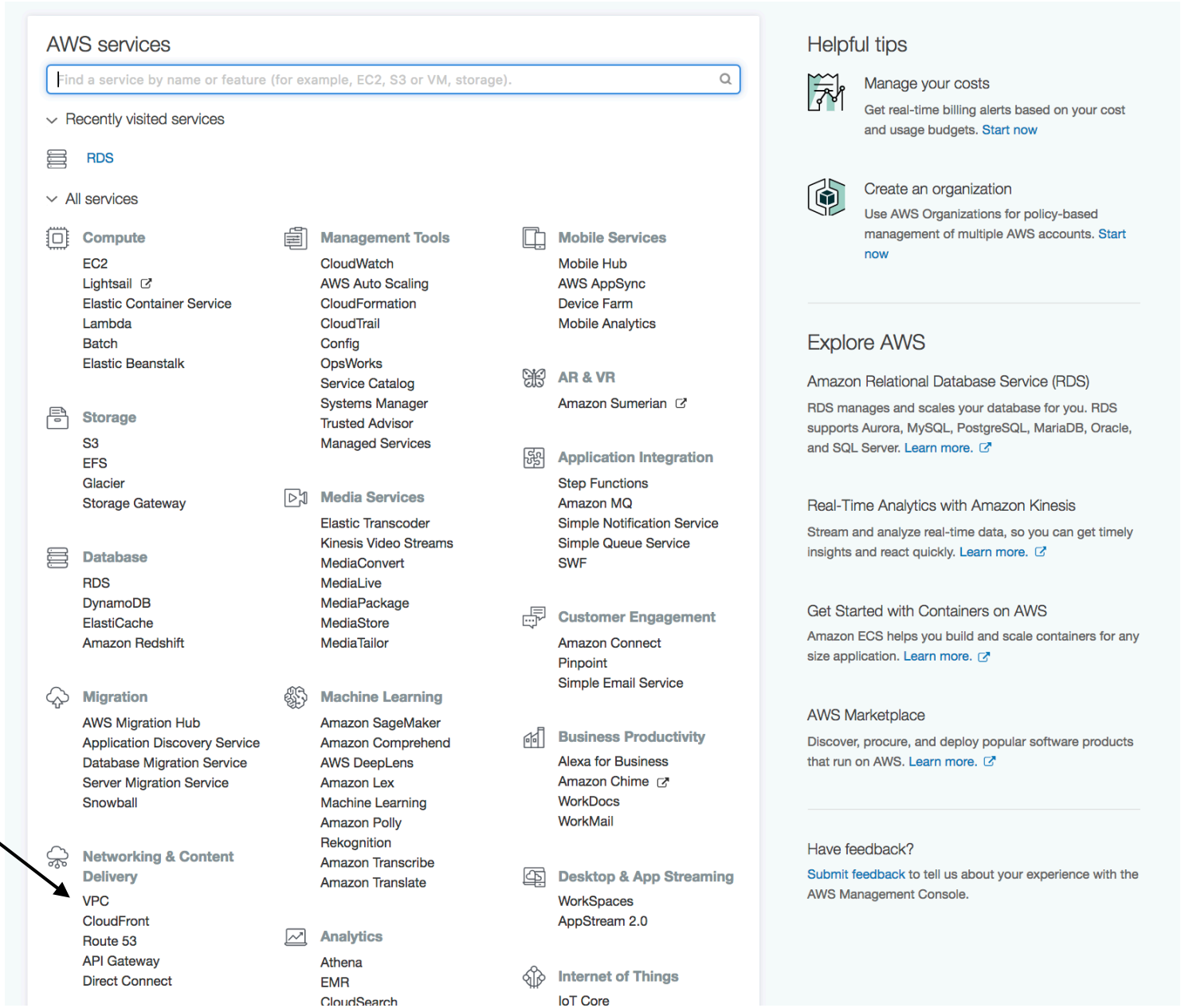
Confirm Publicly Accessible



Security Group

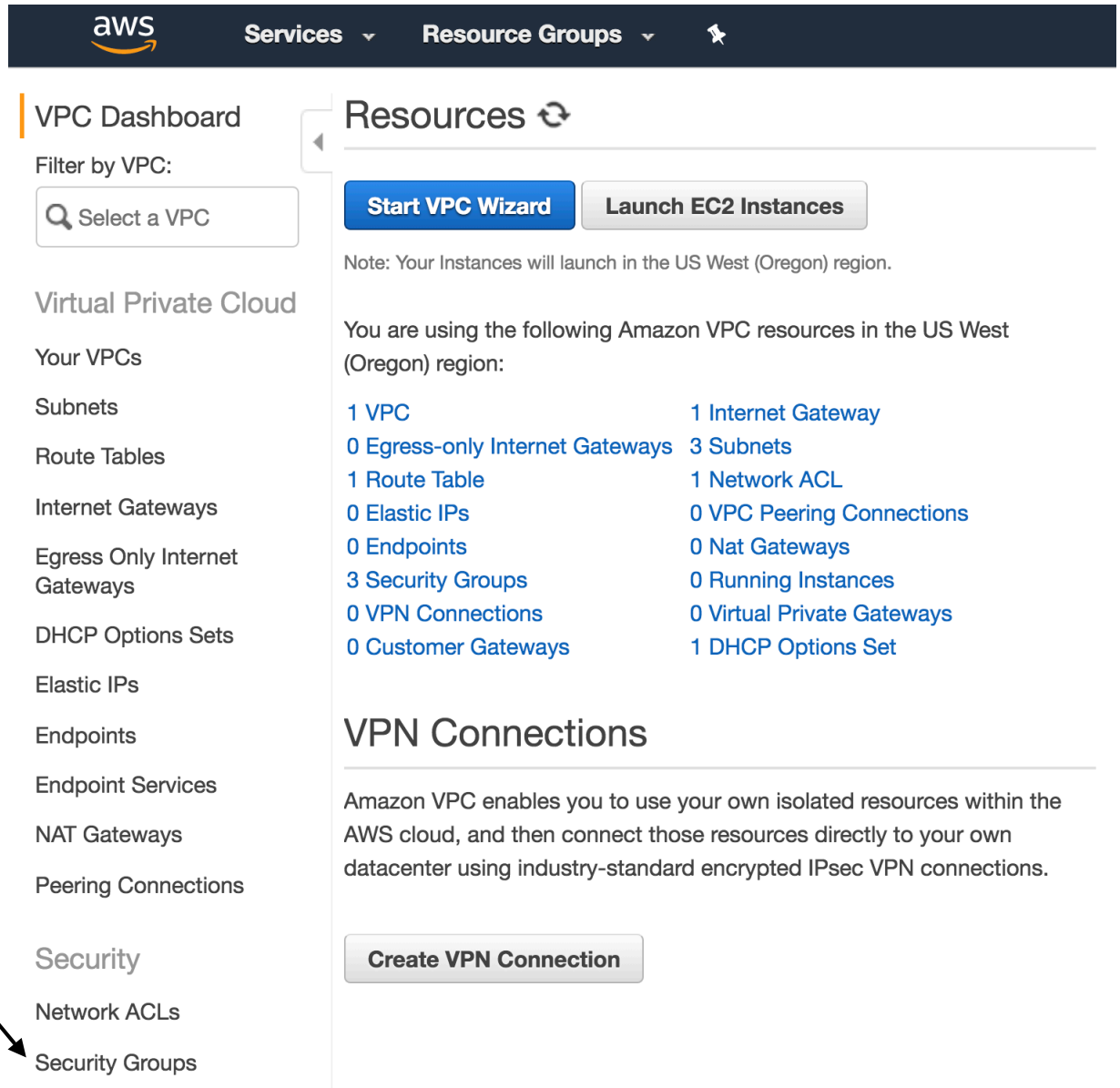
Configure Security

View VPC

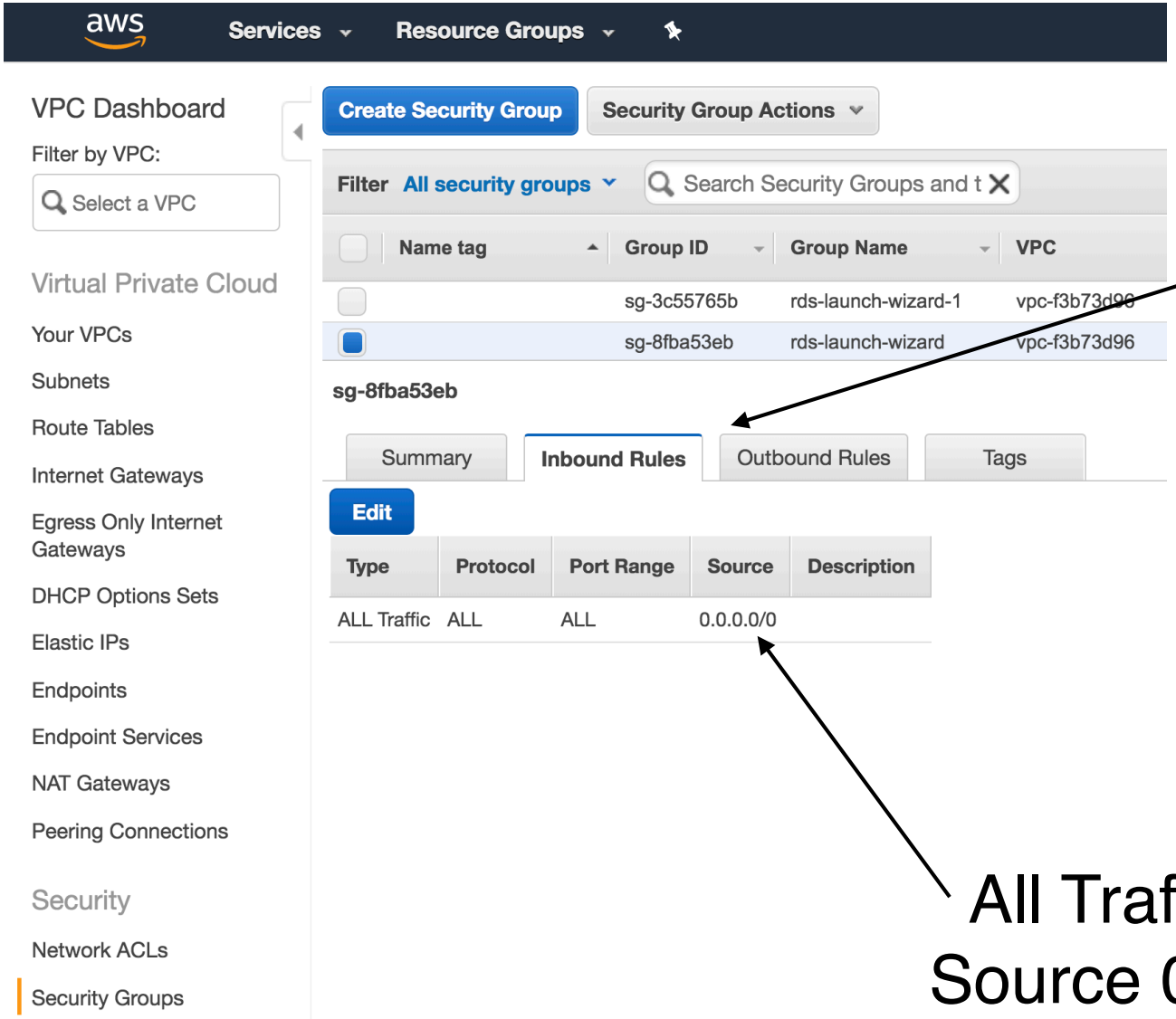


Update Security Groups

View Security Groups



Update Inbound & Outbound Rules



Edit Inbound & Outbound Rules

All Traffic with Source 0.0.0.0/0

QuantSprout

Database Administration

- Create User Credentials
- Grant and Revoke privileges to tables
- Organize database into schemas
- Update database parameters on AWS
  - Increase/decrease instance size
  - Increase memory size (can't decrease)
  - Delete database and save image of data

Create User

- Create new users for your database:

- Syntax:
  CREATE USER <username> WITH PASSWORD <password>;

- Example:
  CREATE USER craig WITH PASSWORD ‘test’;

Grant / Revoke Privileges

Users have no privileges by default

- Syntax:
  GRANT ALL ON <table> TO <user>;
  GRANT SELECT on <table> TO <user>;
  REVOKE SELECT on <table> FROM <user>;

- Example:
  GRANT ALL ON artist TO craig;
  GRANT SELECT ON track TO public;
  REVOKE INSERT ON track FROM public;

Create and Grant Schema

- Syntax:
  CREATE SCHEMA [schema_name];
  GRANT USAGE ON SCHEMA TO [user];

- Example:
  CREATE SCHEMA craig_schema;
  GRANT USAGE ON craig_schema to dylan_moore;

Search Path

- Privileges can be set for each schema
- Reference tables in a schema:
  <schema>.<table>
  public.track
- Set default schema:
  set search_path = <schema1>, <schema2>;
  set search_path = craig, public;
- Everyone has a schema based on the name you used to register (all lower case with spaces replaced with underscore)

Set Your Search Path to Your Schema

Table Options

- Unique – prevents duplicate values
- Not Null – prevents insertion of records with null values for field
- Primary Key – adds unique and non-null constraint
- Foreign Key – adds constraint for field to ensure values exist in corresponding reference table
- Check – applies logical statement as constraint for field
- Serial data type – automatically increments integers for new records

Create Table

- Syntax:
  CREATE TABLE IF NOT EXISTS <table name> (
  <field name1> <data type1>,
  <field name2> <data type2>);

- Example:
  CREATE TABLE IF NOT EXISTS web_log(
  trans_id SERIAL PRIMARY KEY,
  user_name VARCHAR(255),
  transaction VARCHAR(255),
  ts TIMESTAMP)

Import Data

- Load CSV data using WBImport command
- Full documentation at www.sql-workbench.net/manual/command-import.html
- Requires parameters for data loading:
  - Filename and path
  - File type
  - Delimiter
  - Destination table
  - Mapping of data to table fields
  - Date format
  - Batch size (impacts import speed)

Code for Importing Data

```
WbImport
-file='/Users/Craig_Sakuma/Desktop/yelp.csv'
-type=text
-table=public.yelp
-delimiter=|
-timestampFormat='yyyy-MM-dd'
-filecolumns=business_id, review_date,review_id, stars, review_text, review_type, user_id, cool, funny, useful
-batchsize=1000;
```

Amazon Pop-Up Loft

- 1446 Market Street, San Francisco, CA
- aws.amazon.com/start-ups/loft/sf-loft/
- Free help desk and working space
- Host training events

Exercise #4

- Connect to your new AWS databases
- Create a table in your database for the yelp.csv data
- Upload data from the yelp.csv file to the table you've created

yelp.com/biz/quantsprout-san-francisco