

Stage table Schema

The screenshot shows the MySQL Workbench interface with the 'seattle_pet_license2' database selected. The 'stg_seattiestagetable' table is highlighted in the Navigator pane. The main pane displays the table's metadata:

| Name | Engine | Version | Row Format | Rows | Avg Row Length | Data Length | Max Data Length | Index Length | Data Free | Auto Incr |
|-----------------------|--------|---------|------------|-------|----------------|-------------|-----------------|--------------|-----------|-----------|
| stg_seattiestagetable | InnoDB | 10 | Dynamic | 42438 | 111 | 4.5 MB | 0.0 bytes | 0.0 bytes | 4.0 MB | |

The Output pane shows the result of a query: 'SELECT COUNT(*) AS ROW_COUNT FROM stg_seattiestagetable LIMIT 0, 1000'. The result is 1 row(s) returned.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

The screenshot shows the MySQL Workbench interface with the 'seattle_pet_license2' database selected. The 'stg_seattiestagetable' table is highlighted in the Navigator pane. The main pane displays the table's column details:

| Table | Column | Type | Default Value | Nullable | Character Set | Collation | Privileges |
|-----------------------|--------------------|--------------|---------------|----------|---------------|------------------|---------------------------------|
| stg_seattiestagetable | Animal_s_Name | varchar(80) | | YES | utf8mb4 | utf8mb4_0900_... | select,insert,update,references |
| stg_seattiestagetable | DL_CreatedDate | datetime | | YES | utf8mb4 | utf8mb4_0900_... | select,insert,update,references |
| stg_seattiestagetable | DL_ProcessID | varchar(20) | | YES | utf8mb4 | utf8mb4_0900_... | select,insert,update,references |
| stg_seattiestagetable | License_Issue_Date | datetime | | YES | utf8mb4 | utf8mb4_0900_... | select,insert,update,references |
| stg_seattiestagetable | License_Number | varchar(7) | | YES | utf8mb4 | utf8mb4_0900_... | select,insert,update,references |
| stg_seattiestagetable | Primary_Breed | varchar(100) | | YES | utf8mb4 | utf8mb4_0900_... | select,insert,update,references |
| stg_seattiestagetable | Secondary_Breed | varchar(100) | | YES | utf8mb4 | utf8mb4_0900_... | select,insert,update,references |
| stg_seattiestagetable | Species | varchar(20) | | YES | utf8mb4 | utf8mb4_0900_... | select,insert,update,references |
| stg_seattiestagetable | ZIP_Code | varchar(10) | | YES | utf8mb4 | utf8mb4_0900_... | select,insert,update,references |

The Output pane shows the result of a query: 'SELECT COUNT(*) AS ROW_COUNT FROM stg_seattiestagetable LIMIT 0, 1000'. The result is 1 row(s) returned.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Loaded data in stage table

The screenshot shows the MySQL Workbench interface. The 'Query 1' window displays the following SQL script:

```
1 CREATE DATABASE Seattle_Pet_License2;
2
3 USE Seattle_Pet_License2;
4 SELECT* FROM stg_seattlestagetable;
5
6 SELECT COUNT(*) AS ROW_COUNT FROM stg_seattlestagetable;
7
8
```

The 'Result Grid' shows the output of the query. The first table lists pet license data, and the second table shows the row count.

| ZIP_Code | Secondary_Breed | Primary_Breed | Species | Animal_s_Name | License_Number | DI_CreatedDate | DI_ProcessID | License_Issue_Date |
|----------|-----------------|---------------------|---------|---------------|----------------|---------------------|--------------|---------------------|
| 98117 | Mix | Domestic Longhair | Cat | Zen | 5107948 | 2024-02-12 11:27:49 | FebHr | 2015-12-18 00:00:00 |
| 98117 | | Siberian | Cat | Misty | 5116503 | 2024-02-12 11:27:49 | FebHr | 2016-06-14 00:00:00 |
| 98121 | Mix | | Cat | Lyna | 5119301 | 2024-02-12 11:27:49 | FebHr | 2016-08-04 00:00:00 |
| 98107 | | Domestic Longhair | Cat | Veronica | 962273 | 2024-02-12 11:27:49 | FebHr | 2019-02-13 00:00:00 |
| 98115 | | LaPerm | Cat | Spider | 5133113 | 2024-02-12 11:27:49 | FebHr | 2019-08-10 00:00:00 |
| 98125 | | American Shorthair | Cat | Maxx | 8002549 | 2024-02-12 11:27:49 | FebHr | 2019-11-21 00:00:00 |
| 98116 | Mix | Domestic Shorthair | Cat | Sammy | 5151906 | 2024-02-12 11:27:49 | FebHr | 2020-05-19 00:00:00 |
| 98101 | | Siamese | Cat | Bradley | 8015649 | 2024-02-12 11:27:49 | FebHr | 2020-05-19 00:00:00 |
| 98126 | | Domestic Longhair | Cat | Midley | 5142869 | 2024-02-12 11:27:49 | FebHr | 2020-05-24 00:00:00 |
| 98103 | Mix | Domestic Shorthair | Cat | Diamond | 5112835 | 2024-02-12 11:27:49 | FebHr | 2020-07-03 00:00:00 |
| 98119 | | Domestic Shorthair | Cat | Dall | 83454 | 2024-02-12 11:27:49 | FebHr | 2020-07-16 00:00:00 |
| 98126 | Mix | Domestic Shorthair | Cat | Nacho | 511986 | 2024-02-12 11:27:49 | FebHr | 2020-07-21 00:00:00 |
| 98133 | Mix | Domestic Medium ... | Cat | Gracie | 8019541 | 2024-02-12 11:27:49 | FebHr | 2020-08-18 00:00:00 |
| 98116 | Mix | Domestic Shorthair | Cat | Ramon | 5141990 | 2024-02-12 11:27:49 | FebHr | 2020-09-08 00:00:00 |

The 'Action Output' window shows the execution of the query: 'SELECT* FROM stg_seattlestagetable LIMIT 0, 1000'. The message indicates '1000 row(s) returned'.

Talend workflow/job (After run with row counts)

The screenshot shows the Talend Cloud Data Fabric interface. The 'Job LoadSeattlePetLicensesStage' is displayed in the 'Designer' view. The workflow includes the following components:

- SeattlePetLicensesMetadata**: A metadata component.
- tLogRow_1**: A log row component.
- tMap**: A map component.
- MySQLConnection1**: A database connection component.
- tLogRow_2**: A log row component.

The 'Execution' window shows the job status: 'Job LoadSeattlePetLicensesStage ended at 11:27 12/02/2024. [Exit code = 0]'. The 'Statistics' window shows the following data:

| Component | Rows | Time |
|------------------|---------------------|------|
| tLogRow_1 | 0 rows in 2.99s | |
| tMap | 42526 rows in 3.38s | |
| MySQLConnection1 | 12585.38 rows/s | |
| tLogRow_2 | 14241.8 rows/s | |

Count of Rows in the stage table

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'seattle_pet_license2' selected. The main editor shows a SQL query:

```
1 CREATE DATABASE Seattle_Pet_License2;
2
3 USE Seattle_Pet_License2;
4 SELECT* FROM stg_seattlestagetable;
5
6 SELECT COUNT(*) AS ROW_COUNT FROM stg_seattlestagetable;
7
8
```

 The 'Result Grid' shows a single row with the value '42526'. The bottom status bar indicates 'Query Completed'.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

SQL create query for Stage table

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'seattle_pet_license2' selected. The main editor shows a SQL query:

```
1 CREATE DATABASE Seattle_Pet_License2;
2
3 USE Seattle_Pet_License2;
4
5 SELECT* FROM stg_seattlestagetable;
6
7 SELECT COUNT(*) AS ROW_COUNT FROM stg_seattlestagetable;
8
9 SHOW CREATE TABLE stg_seattlestagetable;
```

 The 'Result Grid' shows a single row with the value '42526'. The bottom status bar indicates 'Query Completed'.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.