Connecting, Querying, and Consuming Data

Russ Thomas
DATA ARCHITECT

@sqljudo www.sqljudo.com

Amazon Redshift SQL Syntax

ANSI SQL standards

Redshift quirks

Differences between Redshift and Postgres

- DDL (data definition language)
- DML (data manipulation language)

Connecting Data Consumers

SQL Clients

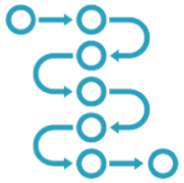
BI Products

ODBC / JDBC Drivers

More demos!

Amazon Redshift SQL Engine

Differences between Redshift and other SQL engines



Supported constraints are for query optimization, not data quality

Some constraints not supported – default, check, exclusions

Amazon Redshift SQL Engine

“Create Index” does not exist

“Alter Column” does not exist

“Alter Table” does support adding and dropping columns

Data Types	Aliases
SMALLINT	INT2
INTEGER	INT, INT4
BIGINT	INT8
DECIMAL	NUMERIC
REAL	FLOAT4
DOUBLE PRECISION	FLOAT8, FLOAT
BOOLEAN	BOOL
CHAR	CHARACTER, NCHAR, BPCHAR
VARCHAR	CHARACTER VARYING, NVARCHAR, TEXT
DATE	
TIMESTAMP	TIMESTAMP WITHOUT TIME ZONE
TIMESTAMPZ	TIMESTAMP WITH TIME ZONE

String and Date Manipulation

CONCAT()
SUBSTRING()
POSITION()
REGEX_*()

+ many more

(search “Redshift String Manipulation”)

DATEADD()
DATEDIFF()
DATE_PART()
LAST_DAY()

+ many more

(search “Redshift Date Manipulation)

Math Functions and Data Conversions

SUM()
ABS()
COS()
FLOOR()
MOD()

+ many more

(Search “Redshift Math Functions”)

CAST()

CONVERT()

(Search “Redshift Cast and Convert”)

Full documentation is found in the *Amazon Redshift Developer Guide*

https://aws.amazon.com/documentation/redshift/

```
BEGIN { TRANSACTION | WORK }
{ISOLATION LEVEL
    SERIALIZABLE
    | READ UNCOMMITTED
    | READ COMMITTED
    | REPEATABLE READ }
. . .
. . .
END |
COMMIT | ROLLBACK
```

- Transaction and work are synonyms
- Only the serializable isolation level is supported, syntax support is to appease ANSI standard

Individual statements outside of BEGIN block are implied transactions with commit, you do not have to specify commit for single statements

```
[ WITH with_subquery [, ...] ]
SELECT [ TOP number | [ ALL | DISTINCT ] * | expression
    [ AS output_name ] [, ...] ]
[ FROM table_reference [, ...] ]
[ WHERE condition ]
[ GROUP BY expression [, ...] ]
[ HAVING condition ]
[ { UNION | ALL | INTERSECT | EXCEPT | MINUS } query ]
[ ORDER BY expression [ ASC | DESC ]
[ LIMIT { number | ALL } ]
[ OFFSET start ]
```

Ranking Window Functions

DENSE_RANK

NTILE

PERCENT_RANK

RANK

ROW_NUMBER

Aggregate Window Functions

AVG	MEDIAN	STDDEV
COUNT	MIN	SUM
CUME_DIST	NTH_VALUE	VAR_POP
FIRST_VALUE	PERCENTILE_CONT	VAR_SAMP
LAG	PERCENTILE_DISC	VARIANCE
LAST_VALUE	RATIO_TO_REPORT	
LEAD	STDDEV_POP	
MAX	STDDEV_SAMP	

Amazon Redshift SQL Syntax

Don’t see a function you like? You can probably still do it

Get familiar with the *Amazon Redshift Database Developer Guide*

Up next: Let’s Get Hands On

Summary

- Drivers and data connectors
- Redshift SQL language
- User defined functions and Python
- Create and replace syntax
- Documentation
- Aliases
- 3rd party functions \ enhancements
- Redshift performance shines bright

Data is valuable!