LAS VEGAS

regoUniversity 2025

Sponsored by ValueOps by Broadcom | Clarity by Broadcom | Rally by Broadcom | ConnectALL by Broadcom | Insights by Broadcom

Basics of SQL

Your Guides:
Shekhar Kanade & David Matzdorf

Gold Sponsor
aws

Introductions

- Take 5 Minutes
- Turn to a Person Near You
- Introduce Yourself

Agenda

- What is SQL
- Relational Databases
- Basic SQL Structure
- How to Query Clarity
- Clarity Data Model
- Write queries

Introduction

What is SQL?

- Structured Query Language
- Standard language to communicate with relational databases
- Query data
  - SELECT
- Manipulate Data
  - DML (Data Manipulation Language)
  - Insert, Update, Delete, Merge
- Defining Data Structures
  - DDL (Data Definition Language)
  - Create, Alter, Drop

What is a Relational Database?

- A collection of tables with pre-defined relationships
- Tables: hold data in rows and columns
  - Columns: represent fields or attributes in a table
  - Rows: One record in a table
- Relationships between tables use a shared attribute (key)
  - Primary Key: Unique Identifier
  - Foreign Key: Link to a primary key in another table
- Vendors: Oracle, Postgres, MS SQL Server, MS Access

Why use Relational Databases?

- Logically organize and store data
- Reduces data duplication
- Keeps information consistent and accurate
- Easy to run powerful queries using SQL

SQL Structure

Basic Structure

- SELECT
  - Tells the database what information you want
  - SELECT id, unique_name, last_name, first_name
- FROM
  - Tells the database what tables to find the data
  - FROM srm_resources
- WHERE
  - Tells the database which rows to include
  - WHERE last_name = ‘Smith’
- ORDER BY
  - Tells the database how to order the results
  - ORDER BY last_name

SELECT Clause

- All Columns
  - SELECT *
- Specific Columns
  - SELECT id, unique_name, last_name, first_name
- Distinct
  - SELECT DISTINCT first_name
- Scalar Subqueries
  - SELECT (SELECT …)
  - Must return only one row and one column

WHERE Clause

- Used to filter records based on specific conditions
- Comparison operators:=, !=, <, <=, >, >=, BETWEEN, IN, LIKE, IS NULL, IS NOT NULL
- Filter By a String: WHERE last_name = ‘Smith’
- Filter By a Number: WHERE id > 23
- Multiple Conditions: WHERE last_name = ‘Smith’ AND first_name = ‘John’
- OR: WHERE last_name = ‘Smith’ OR first_name = ‘John’
- BETWEEN (inclusive): WHERE id BETWEEN 23 AND 27
- IN: WHERE id IN (23, 24, 25, 26, 27)
- LIKE: WHERE last_name Like ‘Smi%’
- NULL: WHERE last_name IS NULL

ORDER BY Clause

- Used to sort the results in a specific order
- Without ORDER BY
  - The database does not guarantee any specific order
  - Order of rows can change between executions of the same query
  - Impacted by physical storage order, indexes, execution plan
- Single Condition: ORDER BY last_name
- Multiple Conditions: ORDER BY last_name, first_name
- Ascending vs Descending
  - Default is Ascending
  - ORDER BY last_name ASC
  - ORDER BY last_name DESC

Basic Query

- SELECT id, unique_name, last_name, first_name
- FROM srm_resources
- WHERE last_name = 'Smith'
- ORDER BY last_name

Table Relationships

- One-to-One (1:1)
  - Each row in Table A matches **exactly one** row in Table B
- One-to-Many (1:N)
  - A row in Table A can match **many** rows in Table B, but rows in Table B match only **one** row in Table A
  - One resource can be assigned to many projects
- Many-to-Many (M:N)
  - Rows in Table A can match **many** rows in Table B, and vice versa
  - Uses a junction table
  - One resource can be assigned to many projects
  - One project can have many resources

JOINS

- Joins are made using keys
  - Primary Keys (PK): Uniquely identifies each row in a table
  - Foreign Keys (FK): A column that refers to a primary key in another table, creating the link between them
- Inner Joins: Returns only the rows where there is a match in both tables
- Left Joins: Returns all rows from the left table, and matching rows from the right. If there’s no match, you get NULLs
- Right Join: Returns all rows from the right table, and matching rows from the left
- Table and Column aliases

INNER JOIN

- SELECT srmr.id
- , srmr.unique_name
- , srmr.last_name
- , srmr.first_name
- , prjr.prisrole
- FROM srm_resources srmr
- JOIN prj_resources prjr ON srmr.id = prjr.prid
- WHERE srmr.last_name = 'Smith'
- ORDER BY srmr.id

LEFT JOIN

- SELECT srmr.id
- , srmr.unique_name
- , tm.prid team_id
- FROM srm_resources srmr
- LEFT JOIN prteam tm ON srmr.id = tm.prresourceid
- WHERE 1=1
- ORDER BY srmr.id

Many-to-Many Example

- SELECT srmr.id
- , srmr.unique_name
- , tm.prid team_id
- , invi.code project_id
- FROM srm_resources srmr
- LEFT JOIN prteam tm ON srmr.id = tm.prresourceid
- LEFT JOIN inv_investments invi ON tm.prprojectid = invi.id
- WHERE 1=1
- ORDER BY srmr.id
- , invi.code

Aggregate Functions

- Takes many rows of data and calculates a single value from them
- Common functions: COUNT, SUM, AVG, MIN, MAX
- Typically used with GROUP BY to get summaries per category
  - Count of project assignments BY resource
  - Sum of allocation hours BY resource BY project
- Filtering Groups (HAVING)
  - Used to filter results after data has been grouped
  - Similar to the WHERE clause but for aggregate results
  - WHERE filters rows before grouping
  - HAVING filters groups after the aggregation is done

Aggregation Example

- SELECT srmr.id
- , srmr.unique_name
- , COUNT(*) cnt
- , SUM(invi.id) sum_of_id
- FROM srm_resources srmr
- JOIN prteam tm ON srmr.id = tm.prresourceid
- JOIN inv_investments invi ON tm.prprojectid = invi.id
- WHERE 1=1
- GROUP BY srmr.id
- , srmr.unique_name
- HAVING COUNT(*) > 5
- ORDER BY srmr.id

Clarity Data Model

Clarity Data Model

- Most data available in Clarity can be easily queried
- Queries are key to customizing Clarity:
  - Dynamic Lookups
  - NSQL Queries/Portlets
  - Custom processes and gel scripting
  - Building Integrations
  - Reporting (Jaspersoft or external systems)

Finding Tables and Columns

- Clarity Studio
- MUX Attributes
- Data Dictionary Tables
  - USER_OBJECTS / SYS.OBJECTS
  - USER_TABLES / INFORMATION_SCHEMA.TABLES
  - USER_TAB_COLUMN / INFORMATION_SCHEMA.COLUMNS
- Rego’s Data Dictionary Extractor
- SQL Trace

How to Query Clarity

- Dynamic Lookup
- Rego HTML Query Portlet
- Japersoft Driver (limited lifespan)
- Query Tool
  - On-prem
  - VPN / AWS lower environments

Core Resource Tables

- SRM_RESOURCES
- PRJ_RESOURCES
- PAC_MNT_RESOURCES
- ODF_CA_RESOURCE

Core Investment Tables

- INV_INVESTMENTS
- PRJ_PROJECTS
- PAC_MNT_PROJECTS
- FIN_FINANCIALS
- ODF tables

Investment Sub-Object Tables

- PRTEAM
- PRTASK
- PRASSIGNMENT
- ODF_CA_COP_PRJ_STATUSRPT / COP_PRJ_STATUSRPT_LATEST_V
- RIM_RISK_AND_ISSUES
- PRJ_BASELINES
- PRJ_BASELINE_DETAILSPRJ_BASELINES
- PRJ_BASELINE_DETAILS

Cost Plan Tables

- Plan Table: FIN_PLANS
  - PLAN_TYPE_CODE
- Cost Plan Details
  - FIN_COST_PLAN_DETAILS
  - ODF_SSL_CST_DTL_UNITS
  - ODF_SSL_CST_DTL_COST
- Benefit Plan Details
  - FIN_BENEFIT_PLAN_DETAILS
  - ODF_SSL_BFT_DTL_BFT

Timesheet Tables

- PRTIMEPERIOD
- PRTIMESHEET
- PRTIMEENTRY

regoUniversity

Master Clarity with Rego University

Earn Certifications in Administration, Leadership, and Technical Proficiency

Let Rego be your guide.

Elevate Your Professional Expertise with Rego University Certifications

Rego is excited to continue our **certification programs**, designed to enhance your expertise in Clarity administration, leadership, and technical skills. These certifications provide hands-on experience and knowledge to excel in your career.

Certification Requirements:

- ✓ **Completion**: 12 units per certification track
- ✓ **Eligibility**: Open to all Rego University attendees

Important Reminder:

To have your certification **credits tracked**, ensure you **complete the class surveys in the app** after each session. This step is critical for certification progress.

Questions?

Surveys

Please take a few moments to fill out the class survey.
Your feedback is extremely important for future events.

Thank You For Attending Rego University

Instructions for PMI credits

- Access your account at pmi.org
- Click on **Certifications**
- Click on **Maintain My Certification**
- Click on **Visit CCR's** button under the **Report PDU's**
- Click on **Report PDU's**
- Click on **Course or Training**
- Class Provider = **Rego Consulting**
- Class Name = **regoUniversity**
- Course **Description**
- Date Started = **Today's Date**
- Date Completed = **Today's Date**
- Hours Completed = **1 PDU per hour of class time**
- Training classes = **Technical**
- Click on **I agree** and **Submit**

Let us know how we can improve!
Don't forget to fill out the class survey.

Phone
888.813.0444

Email
info@regoconsulting.com

Website
www.regouniversity.com

Continue to Get Resources and Stay Connected

1. Use RegoXchange.com for instructions and how-tos.
2. Talk with your account managers and your Rego consultants.
3. Connect with each other and Clarity experts at RegoGroups.com.
4. Sign up for webinars and join in-person Rego groups near you through at RegoConsulting.com
5. Join us for the next Rego University!