

SQL Exercises

Bruce Momjian

February 2004

1. Display psql's SQL help and command help outputs
2. Create a table; INSERT, UPDATE, and DELETE using it
3. Create a SELECT that has spaces in a column label

4. Insert a single quote into a text field
5. Test for NULLs
6. Use a WHERE and ORDER BY clauses with the above queries
7. INSERT and SELECT NULL values
8. Use DEFAULT values

9. Use LIKE, ILIKE (case insensitive), and regular expressions in queries

10. Test NULL handling of aggregates
11. Use GROUP BY and HAVING

12. Add additional entries to the order entry test database
13. Add a COMMENT to order entry tables

14. Does setval() lock the counter inside a transaction?
15. Create a gap in a SERIAL column

16. Try UNION, INTERSECT, and EXCEPT

17. Use subqueries
18. Use CREATE TABLE AS to make a copy of a system table
19. Use INSERT ... SELECT to repeatedly double the number of rows in a table

20. Try some of the psql \f functions
21. Illustrate the difference between CURRENT_TIMESTAMP and timeofday()
22. Experiment with TIMESTAMP with and without timezone

23. Cause a deadlock
24. Cause a query to block by updating the same row at the same time
25. Cause a query to block by using SELECT ... FOR UPDATE

26. Cause a query to block using LOCK
27. Show different output with READ COMMITTED vs SERIALIZABLE ISOLATION LEVEL
28. Use LOCK TABLE to simulate sequences

29. Use LIMIT in a query
30. Create a temporary table
31. Create a view
32. Create a view with rules that behaves like an ordinary table in terms of SELECT, INSERT, UPDATE, and DELETE. Find a way in which it doesn't behave like an ordinary table.

33. Test NOT NULL constraint
34. Test UNIQUE constraint on one column. On two columns.
35. Create a PRIMARY KEY in a table
36. Reference it from another table with a REFERENCES foreign key
37. Show a referential integrity failure
38. Show a case of a NO ACTION constraint. CASCADE constraint. SET NULL constraint. SET DEFAULT constraint.
39. Create a CHECK constraint for even numbers

40. COPY a table out, modify the data with a text editor, and reload into a different table
41. Convert a quoted, comma-separated value file into a format acceptable to COPY
42. Use '|' as a delimiter and COPY '|' into a text column

43. Create a server-side SQL function
44. Create an SQL function to add shipping costs based on the following formula: <= \$25, \$5, > \$25 and <= \$50, \$10, >\$50, \$15. Create a PL/PgSQL function to do the same thing.
45. Load the PL/pgSQL language into the database
46. Create a PL/pgSQL function
47. Create a PL/pgSQL function to constrain a column's value
48. Use the function as a DEFAULT for a column

49. *Load all the /usr/bin files into the database as large objects
50. *Load all the /usr/bin files into the database in a bytea column
51. *Create a C function that loads into the database

52. Use ANSI-style join syntax
53. Show the difference between FULL OUTER, LEFT OUTER, and RIGHT OUTER
54. Show a CROSS JOIN. Show a cartesian product.
55. How is query output different when joining with USING compared to ON?
56. Create a user-specific schema and create the same table in that and the public schema. Access both in the same query.