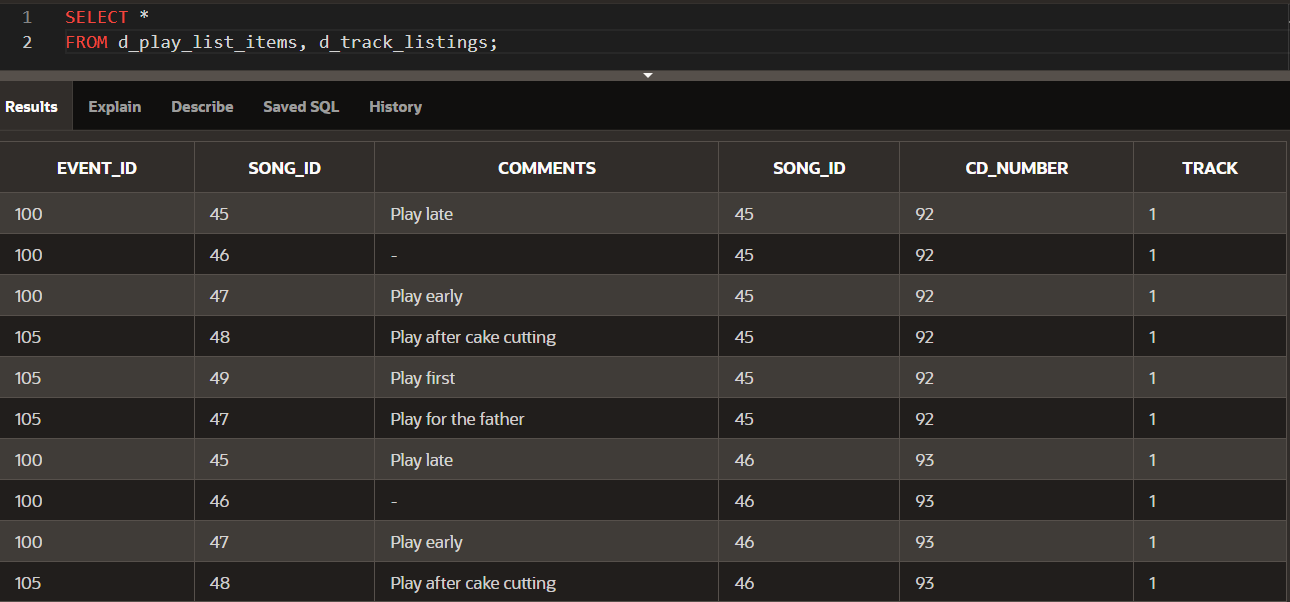
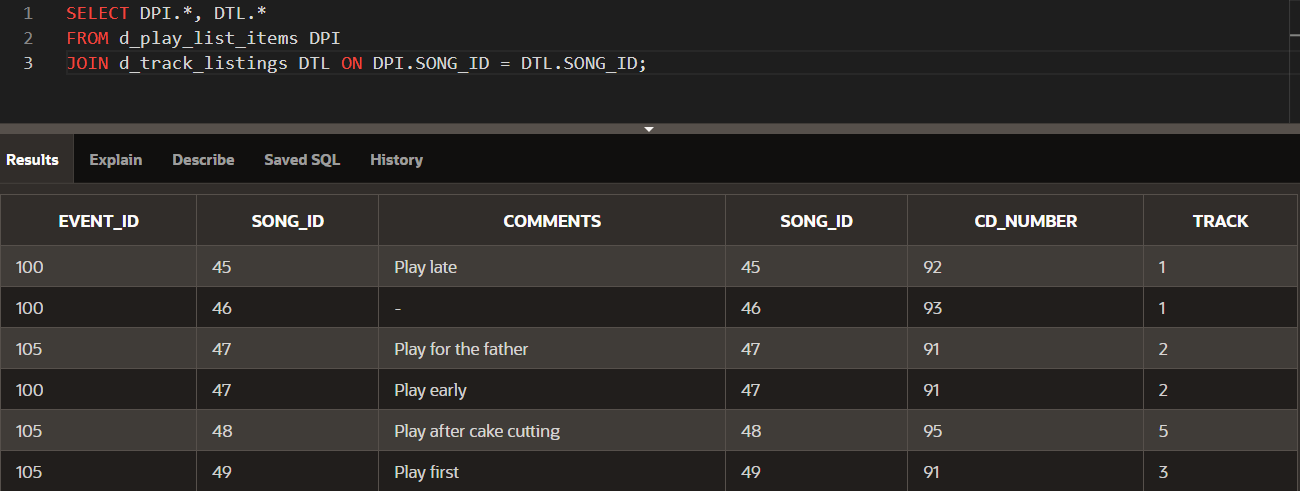
# Database Programming with SQL

**7-1: Oracle Equijoin and Cartesian Product**

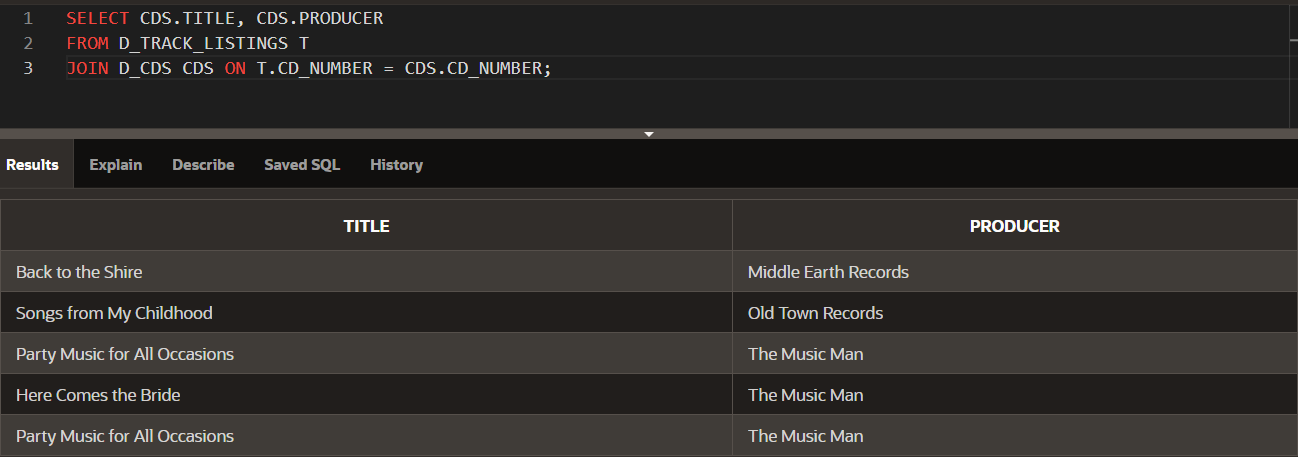
1. Create a Cartesian product that displays the columns in the d\_play\_list\_items and the d\_track\_listings in the DJs on Demand database.



1. Correct the Cartesian product produced in question 1 by creating an equijoin using a common column.



1. Write a query to display the title, type, description, and artist from the DJs on Demand database.

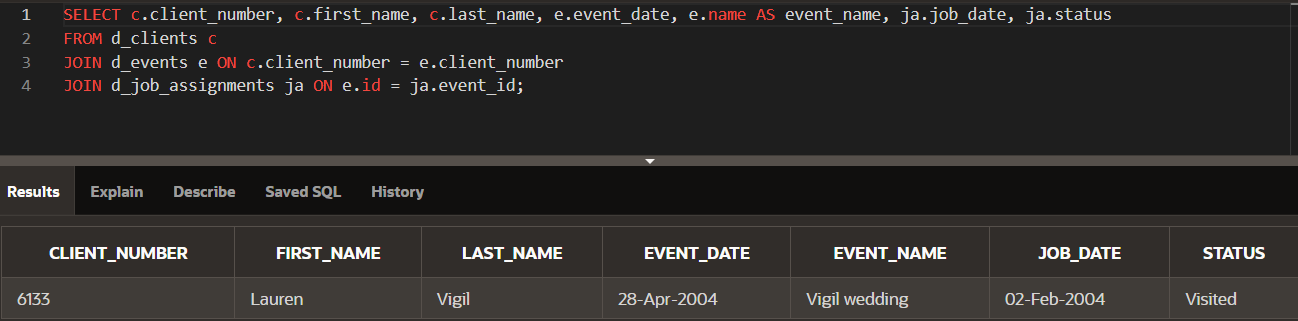


The D\_CDS table doesn’t have columns type and description, so only title and producer have been included in this query.

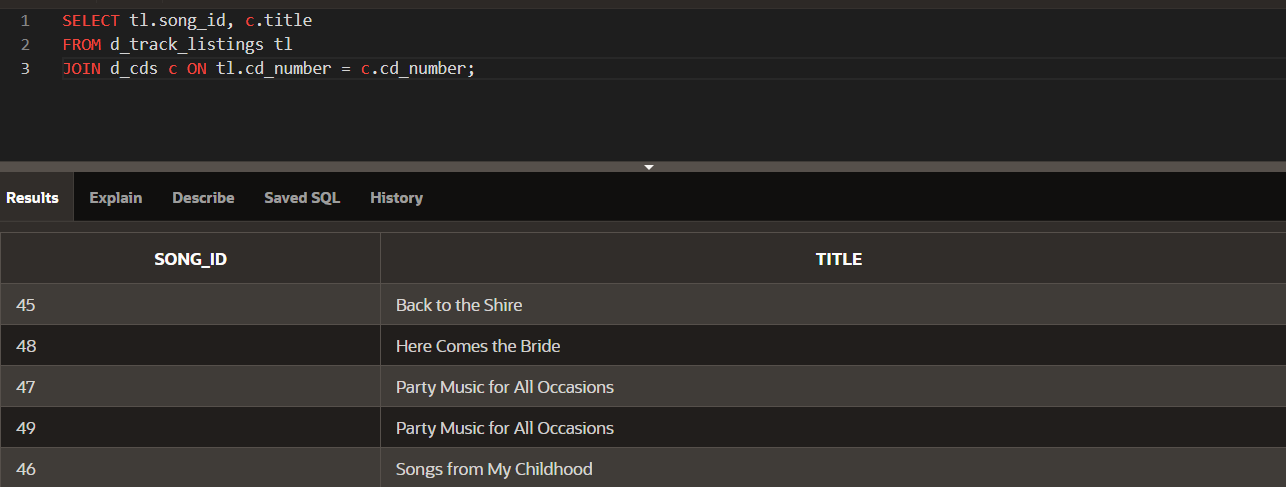
1. Rewrite the query in question 3 to select only those titles with an ID of 47 or 48.

No title with ID’s 47 and 48 exist.

1. Write a query that extracts information from three tables in the DJs on Demand database, the d\_clients table, the d\_events table, and the d\_job\_assignments table.



1. Create and execute an equijoin between DJs on Demand tables d\_track\_listings and d\_cds. Return the song\_id and the title only.



1. Mark T for the statements that are true and F for the statements that are false.

\_\_**T**\_\_ a. A join is a type of query that gets data from more than one table based on columns with the same name.

\_\_**T**\_\_ b. To join tables using an equijoin, there must be a common column in both tables and that column is usually a primary key in one of the tables.

\_\_**T**\_\_ c. A Cartesian product occurs because the query does not specify a WHERE clause.

\_\_**F**\_\_ d. Table aliases are required to create a join condition.

\_\_**T**\_\_ e. If a table alias is used for a table name in the FROM clause, it must be substituted for the table name throughout the SELECT statement.

\_\_**F**\_\_ f. Table alias must be only one character in length.

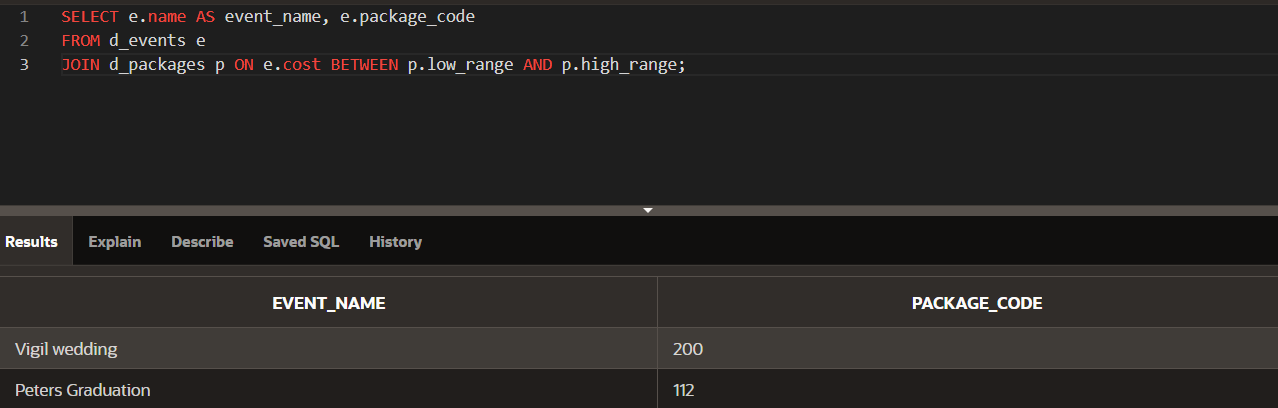
\_\_**T**\_\_ g. A simple join or inner join is the same as an equijoin.

1. What advantage does being able to combine data from multiple tables have for a business?

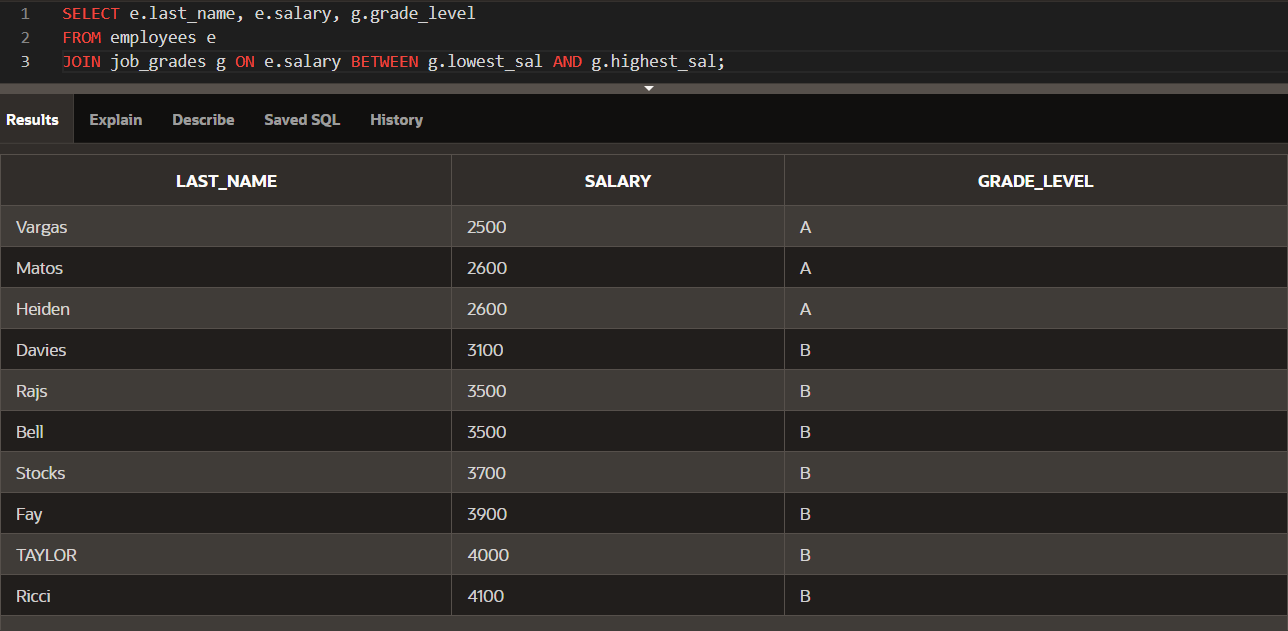
* **Comprehensive Insights**: Businesses can gain a more holistic view of their operations, customers, and events by combining data from different sources.
* **Enhanced Decision-Making**: By analyzing data across multiple tables, businesses can make more informed decisions, leading to better strategies and outcomes.
* **Efficiency in Data Management**: Combining data reduces redundancy and improves data integrity, leading to a more streamlined database management process.
* **Customized Reports**: Businesses can create detailed reports that cater to specific needs by merging relevant data from various tables.
* **Improved Customer Service**: By analyzing customer interactions and event details together, businesses can enhance their service offerings and customer satisfaction.

**7-1: Oracle Equijoin and Cartesian Product**

1. Create a join based on the cost of the event between the DJs on Demand tables D\_EVENTS and D\_PACKAGES. Show the name of the event and the code for each event.

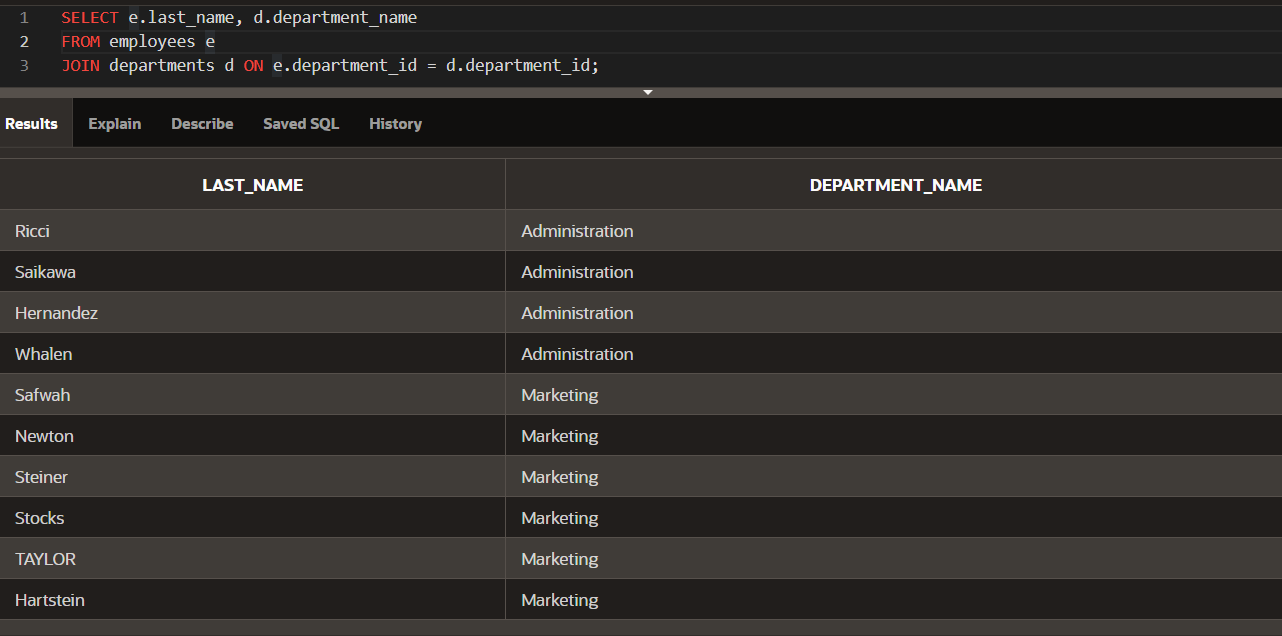


1. Using the Oracle database, create a query that returns the employee last name, salary, and job\_grade level based on the salary. Select the salary between the lowest and highest salaries.



1. What condition requires the creation of a nonequijoin?
   * + When equality isn’t involved between two tables during join condition, a **nonequijoin** is required. For example, when comparing ranges, a nonequijoin is used.
2. Rewrite the following nonequijoin statement using the logical condition operators (AND, OR, NOT): WHERE a.ranking BETWEEN g.lowest\_rank AND g.highest\_rank
   * + WHERE a.ranking >= g.lowest\_rank AND a.ranking <= g.highest\_rank;
3. How do you know when to use a table alias and when not to use a table alias?
   * + **Use a table alias** when:
       - You are joining multiple tables to avoid ambiguity between column names.
       - You want to shorten the table name for readability.

Example:



* + - Do not use a table alias when:
      * There is only one table or when aliases would not improve the clarity of your SQL.

1. What kind of join would you use if you wanted to find data between a range of numbers?
   * + To find data between a range of numbers, a **nonequijoin** is appropriate. This is because the condition for joining the tables is based on a range rather than equality.
2. You need to produce a report for Global Fast Foods showing customers and orders. A customer must be included on the report even if the customer has had no orders.

