

Using the Analysis tab

The **Analysis** tab provides detailed information about statements and cases selected from the **Overview** tab, after a tuning job has been executed. It also shows filter ratio, and table and join sizes.

The **Analysis** tab contains information about the statement or case, its full SQL code, a diagram of the SQL statement, and Index Analysis.

The screenshot shows the Oracle SQL Analysis interface. The top bar indicates the current database is 'TORLABSCORCL'. The 'Analysis' tab is selected. The 'SQL Analysis' section shows the selected statement of interest: 'SELECT 1' with a rewrite of '- [IN_TO_EXISTS] SQLRewrite'. The SQL query is displayed on the left, and the execution plan is shown on the right. The execution plan includes a subquery (1) joined to MOVIERENTAL (MR), which is joined to CUSTOMER (CS). MOVIERENTAL (MR) is also joined to RENTALITEM (RI), which is joined to NOT IN (2). The bottom section, 'Collect and create indexes', shows a table of index recommendations.

Index Name	Table Owner	Table Name	Index Type
IDX_MOVIERENTAL_0	MOVIES	MOVIERENTAL	TOTALCH
CUSTOMER_PK	MOVIES	CUSTOMER	CUSTOMER
MOVIECOPY_PK	MOVIES	MOVIECOPY	MOVIECO
RENTALITEM_FK1	MOVIES	RENTALITEM	RENTALIC
CUSTOMER_IJ1	MOVIES	CUSTOMER	LASTNAMI

Additionally, for the Oracle and SQL Server platforms there are Table Statistics, Column Statistics and Histograms, and Outlines/Plan Guides tabs. For more information, see [Using platform-specific features](#).

Statement analysis is performed when you click **Perform detail analysis** on the **Overview** tab and then click **Run Job** or when you click the **Analysis** tab. In order to view and analyze statement statistics, select the tab (Index Analysis, Table Statistics, Column Statistics and Histograms, or Outline) and the statements whose statistics you want to analyze.

Next to the **Select statement of interest** list at the top, you choose to see an analysis of the **>ROOT** statement, or you can click the list and see an analysis of any one of the generated cases produced by running the tuning job from the **Overview** tab.

For more information, see [Visual SQL tuning](#).

Implementing index analysis recommendations

Once you have added tuning candidates to a tuning job, DB Optimizer can analyze the effectiveness of the indexes in the database and recommend the creation of new indexes where the new indexes can increase performance.

In the **Collect and create indexes** table, any indexes DB Optimizer recommends you create are marked in orange.

Index Analysis | Table Statistics | Column Statistics And Histograms | Outlines

Collect and create indexes

Index Name	Table Owner	Table Name	Column Name	Index Type
<input checked="" type="checkbox"/> IDX_MOVIERENTAL_0	MOVIES	MOVIERENTAL	TOTALCHARGE	Normal
<input checked="" type="checkbox"/> CUSTOMER_PK	MOVIES	CUSTOMER	CUSTOMERID	Unique
<input checked="" type="checkbox"/> MOVIECOPY_FK	MOVIES	MOVIECOPY	MOVIECOPYID	Unique
<input checked="" type="checkbox"/> RENTALITEM_FK1	MOVIES	RENTALITEM	RENTALID	Normal
<input type="checkbox"/> CUSTOMER_IJEL	MOVIES	CUSTOMER	LASTNAME	Normal

Table MOVIES.MOVIERENTAL is scanned via full table scan but it has a filter `mv.totalcharge > (select avg (totalcharge) from MOVIES.movierental)` on it and we created a virtual index `IDX_MOVIERENTAL_0` which the optimizer picked up, so we suggest implementing

Create Index

To accept the suggestion and have tuning automatically generate an index

1. For any recommended index, click the checkbox to the left of the index.
Optionally, modify the Index type by clicking in the **Index Type** column and then selecting a type from the list.
2. Click the **Create Indexes** button.
The **Index Analysis** dialog appears.

To view the index SQL in an editor for later implementation, click the statement and then click **Open in a SQL editor**.

To run the index SQL and create the index on the selected database, click **Execute**.