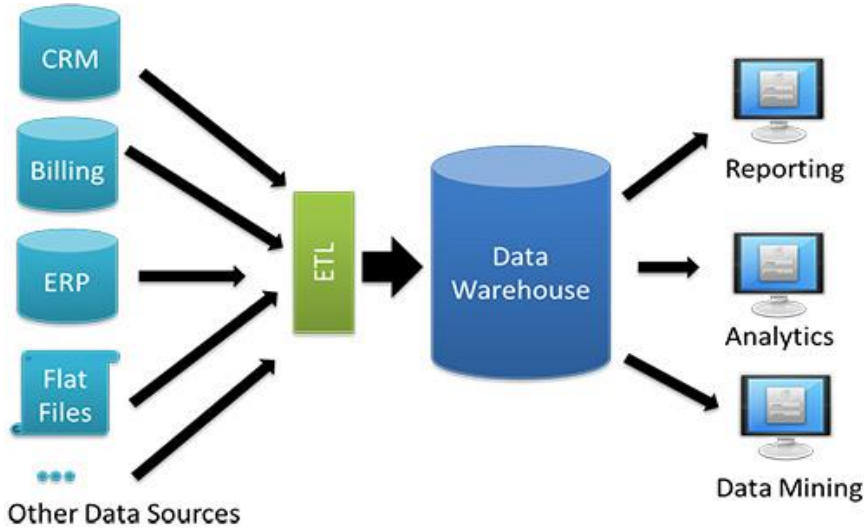
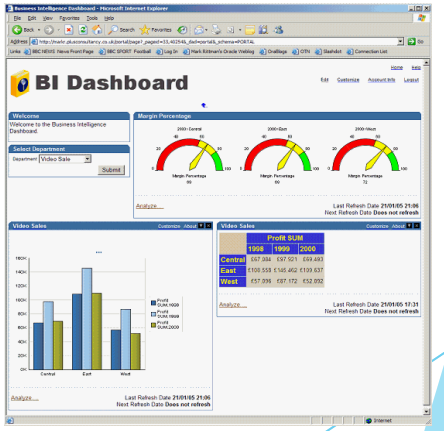
Manipulating Data

What is a Data Warehouse?

A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing.

It usually contains historical data derived from transaction data, but it can include data from other sources. It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources.
In addition to a relational database, a data warehouse environment includes an extraction, transportation, transformation, and loading (ETL) solution, an online analytical processing (OLAP) engine, client analysis tools, and other applications that manage the process of gathering data and delivering it to business users.



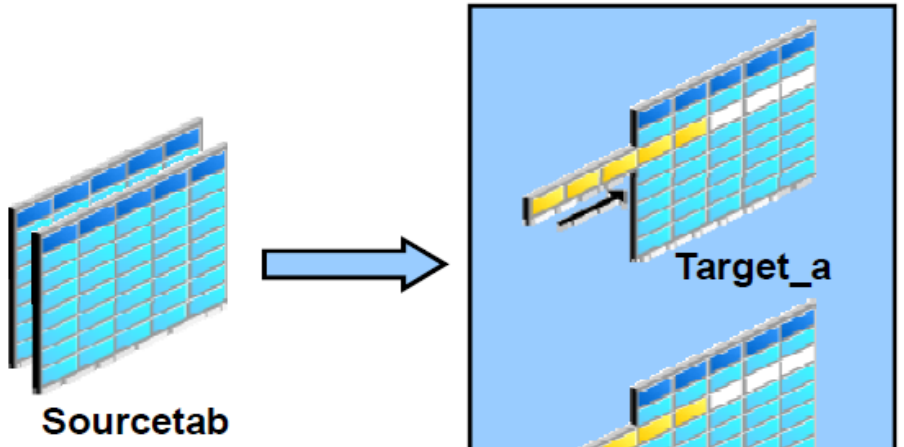


Agenda

- Using the explicit default value in insert or update statement.
- Copy rows from another table.

- Using the following types of multitable `INSERT`s:
  - Unconditional `INSERT`
  - Conditional `INSERT ALL`
  - Conditional `INSERT FIRST`
  - Pivoting `INSERT`
- Merging rows in a table
- Performing flashback operations
- Tracking the changes to data over a period of time

Multitable INSERT Statements: Overview




```
INSERT  ALL
   INTO target_a VALUES(…,…,…)
   INTO target_b VALUES(…,…,…)
   INTO target_c VALUES(…,…,…)
   SELECT …
   FROM  sourcetab
   WHERE …;
```

Subquery

Multitable INSERT Statements: Overview

- Use the INSERT...SELECT statement to insert rows into multiple tables as part of a single DML statement.
- Multitable INSERT statements are used in data warehousing systems to transfer data from one or more operational sources to a set of target tables.
- They provide significant performance improvement over:
  - Single DML versus multiple INSERT...SELECT statements
  - Single DML versus a procedure to perform multiple inserts by using the IF...THEN syntax

You use different clauses to indicate the type of INSERT to be executed. The types of multitable INSERT statements are:

- **Unconditional INSERT:** For each row returned by the subquery, a row is inserted into each of the target tables.
- **Conditional INSERT ALL:** For each row returned by the subquery, a row is inserted into each target table if the specified condition is met.
- **Conditional INSERT FIRST:** For each row returned by the subquery, a row is inserted into the very first target table in which the condition is met.
- **Pivoting INSERT:** This is a special case of the unconditional INSERT ALL.

Restrictions on Multitable INSERT Statements

- You can perform multitable INSERT statements only on tables, and not on views or materialized views.
- You cannot perform a multitable INSERT on a remote table.
- You cannot specify a table collection expression when performing a multitable INSERT.
- In a multitable INSERT, all insert_into_clauses cannot combine to specify more than 999 target columns.

MERGE Statement Syntax

You can conditionally insert, update, or delete rows in a table by using the MERGE statement.

```
MERGE INTO table_name table_alias
  USING (table|view|sub_query) alias
  ON (join condition)
  WHEN MATCHED THEN
    UPDATE SET
    col1 = col1_val,
    col2 = col2_val
 WHEN NOT MATCHED THEN
    INSERT (column_list)
    VALUES (column_values);
```

Table A

ID	name
1	khaled
2	ali
3	ahmed

Table B

ID	name
1	xxxxx
2	xxxxx

```
merge into table_b b
using (select * from table_a) a
on ( b.id=a.id)
when matched then
update
set b.name=a.name
when not matched then
insert values (a.id, a.name );
```

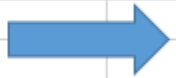


Table B

ID	name
1	khaled
2	ali
3	ahmed

FLASHBACK TABLE Statement

- Enables you to recover tables to a specified point in time with a single statement
- Restores table data along with associated indexes and constraints
- Enables you to revert the table and its contents to a certain point in time or system change number (SCN)

Oracle Flashback Table enables you to recover tables to a specified point in time with a single statement. You can restore table data along with associated indexes and constraints while the database is online, undoing changes to only the specified tables.

The Flashback Table feature is similar to a self-service repair tool. For example, if a user accidentally deletes important rows from a table and then wants to recover the deleted rows, you can use the `FLASHBACK TABLE` statement to restore the table to the time before the deletion and see the missing rows in the table.

When using the `FLASHBACK TABLE` statement, you can revert the table and its contents to a certain time or to an SCN.

Note: The SCN is an integer value associated with each change to the database. It is a unique incremental number in the database. Every time you commit a transaction, a new SCN is recorded.

Thank You