Data Warehousing with Oracle

Comprehensive Concepts Overview, Insight, Recommendations, Best Practices and a whole lot more.

By Tariq Farooq
A “BrainSurface” Presentation
What is a Data Warehouse?

- Designed and built for Performance, Performance & Performance!
- Relational database designed for query and analysis instead of transaction processing.
- Typically contains historical data.
- Consolidates data from various sources.
- Separates Analysis Workload from Transaction Workload.
- Has an ETL (Extraction, Transformation & Loading) engine.
What is a Data Warehouse?

- Designed for OLAP, Data Mining, Client Analysis Tools and other Data Gathering / Analysis applications.
- Good Design includes a Staging Area for receiving / transforming data from various transactional processing / feeder systems.
- Has Metadata, Raw Data, Mining Data & Summary Data.
- May contain Data Marts (Categorized subsets of a data warehouse).
Characteristics of a Data Warehouse

According to William Inmon, a Data Warehouse has the following main characteristics:

- Subject Oriented
- Integrated
- Non-volatile
- Time Variant
Comparison of OLTP and Data Warehouse Systems

Data Warehouse Design

- Architected and designed for ad-hoc queries and analysis.
- DML: Gets data from Transactional Processing / Feeder systems (mostly in bulk-loads).
- Schemas are de-normalized.
- Large volumes of data (thousands, millions, billions of records) are read for analysis.
- Contains Archived/Historical Data.
Data Warehouse Architecture

- Transactional Processing Systems feed data to the Data Warehouse.
- Staging Area is used to Cleanse and Process (Transform) your data.
- Pre-computed Aggregates are stored in Summary Data.
- Data Warehouse can contain Data Marts which are separated and categorized subsets of a Data Warehouse.
Benefits of Data Warehouses

- Massive Information Storage & Retrieval for reporting, analysis & decision-making.
- Consolidation of Information from various (similar and disparate) sources.
- Archival of historical Enterprise data.
- Enterprise-wide (Company-as-a-whole) Information is available for making well-informed, intelligent and time-sensitive decisions: Therefore, Data Warehouses are also called Decision Support Systems (DSS).
Information Retrieval from Data Warehouses

- Reporting
- Data Mining
- Data Visualization
- Online Analytical Processing (OLAP)
Designing a Data Warehouse

- Logical Design
- Physical Design
Data Warehouse: Logical Design

- Identify Business Requirements
- Create a Conceptual Design
- Use Entity-Relationship Diagramming
Data Warehouse: Logical Design

- Translate and map your design into dimensional data warehouse schemas:
  - Star Schemas
  - Snow-flake Schemas
Data Warehouse: Logical Design: What is a Star Schema?

- Looks like a star.
- Most commonly used dimensional model in Data Warehouses.
- Recommended for its simplicity, ease-of-use and most importantly performance.
- Entity Relationships One-Level deep only.
Data Warehouse: Logical Design: What is a Star Schema?

- Has one OR more fact tables.
- Each fact table has one OR more Dimension Tables joined to it.
- Fact tables are huge.
- Dimension tables are small.
- Referential Integrity constraints defined between Fact and Dimension tables.

Star Schema consists of Facts and Dimension tables

Diagram/Figure from Oracle Documentation
Data Warehouse: Logical Design: Dimensions

- Dimensions categorize data.
- Dimensions have one OR more hierarchies.
- Hierarchies are ordered levels of categorizing data within a dimension.
- Hierarchies can be used for data aggregation with different levels of granularity.

Example of a Dimension Hierarchy

Diagram/Figure from Oracle Documentation
Data Warehouse: Physical Design

- Translate and Map Logical Design into Physical Design:
  - Tablespaces
  - Tables
  - Partitions on Tables
  - Indexes
  - Materialized Views
  - Integrity Constraints
  - Dimensions

Logical Design vs. Physical Design

Diagram/Figure from Oracle Documentation
Data Warehouse: Physical Design: Systems Planning

- Recommendation: Plan for high Bandwith on I/O.
- Recommendation: Use parallelization on Database Instances (Oracle Real Application Clusters) to provide load-balancing and automatic failover for business continuity.
- Recommendation: Stripe and Mirror Everything (SAME).
- Storage Capacity Planning.
Recommendation: Partition your large tables into smaller manageable chunks.

Partition Pruning helps queries scan for data in relevant partitions thereby reducing overall logical cost significantly.

Recommendation: Composite Partitioning combines partitioning methods for faster performance e.g. Range-List, Range-Hash.
Data Warehouse: Physical Design: Indexes

- Recommendation: Use Bitmap Indexes in Data Warehouses.
- Bitmap Indexes are meant for columns with low cardinality.
- Bitmap Indexes significantly improve performance for ad-hoc queries.
- Recommendation: Bitmap Indexes on partitioned tables must be local.
Data Warehouse: Physical Design: Integrity Constraints

- Recommendation: Avoid performance overhead generated because of maintaining / enforcing constraints in Data Warehouses.
- Recommendation: Foreign Key Constraints: Use ENABLE NOVALIDATE clause.
- Recommendation: Because Transactional Processing Systems and ETL Processes regularly validate data integrity, RELY (Belief / Non-Enforceable) clause can be used as an alternative to above e.g. RELY DISABLE NOVALIDATE.
Materialized views are used to store pre-computed summaries / aggregates of data in Data Warehouses.

Recommendation: Create Fast-Refreshable Materialized views using Materialized View Logs.

Materialized Views enable QUERY_REWRITE, which results in amazing performance gains.
Recommendation: Define Materialized Views and Dimensions.


Alter System SET QUERY_REWRITE_ENABLED = TRUE.
Data Warehouse: Physical Design: Materialized Views

- Transparent Query Rewrite dramatically increases query speed.
- Query Optimizer automatically re-routes the query to the materialized view for better performance.
- Materialized Views can be partitioned for performance.

Diagram/Figure from Oracle Documentation
Partition Change Tracking identifies which rows in a materialized view are affected by a detail table partition.

Partition Change Tracking enables PCT refresh on Materialized Views for tracking a finer grain of freshness.

DBMS_MVIEW.PMARKER function significantly reduces the cardinality of the materialized view.
Data Warehouse: Physical Design: Extraction, Transformation & Loading (ETL)

- Data Extraction from various (homogenous/heterogeneous) sources.
- Physical Transportation of potentially large volume of data from source systems to target warehouse.
- Transformation and Loading of data into Target Warehouse.
Data Warehouse: Physical Design: Extraction, Transformation & Loading (ETL)

- Types of Extraction
  - Full Extraction
  - Logical Extraction

- Extraction Methods
  - Online Extraction
  - Offline Extraction
Data Warehouse: Physical Design: Extraction, Transformation & Loading (ETL)

- Online Extraction
  - Direct Connection to source system
  - Snapshot Logs, Change Tables etc.

- Offline Extraction
  - Dump/Flat Files
  - Redo/Archive Logs
  - External Tables
  - Transportable Tablespaces
Data Warehouse: Physical Design: Extraction, Transformation & Loading (ETL)

- Incremental Extraction: Change Data Capture
  - Tremendously Efficient
  - Trickle-Feeding Data Acquisition & Delivery
  - Enables Near-Real-Time Data Warehousing
  - Can be implemented by the following methods:
    - Triggers
    - Timestamps
    - Partitioning
Data Warehouse: Physical Design: Extraction, Transformation & Loading (ETL)

- Transportation Mechanisms:
  - Flat Files
  - Distributed Query Operations
  - Transportable Tablespaces

- Copy generated data using periodically run batch processes.
Data Warehouse: Physical Design: Extraction, Transformation & Loading (ETL)

- Loading & Transformation:
  - Complex & Costly.
  - Can range from simple data conversions to highly complex methods.

- Loading Mechanisms:
  - SQL*Loader.
  - External Tables.
  - Transportable Tablespaces.
  - Custom Code/Scripts/APIs.
Data Warehouse: Physical Design: Extraction, Transformation & Loading (ETL)

Flow of a Pipelined Data Transformation

Diagram/Figure from Oracle Documentation
Data Warehouse: Physical Design: Extraction, Transformation & Loading (ETL)

Flow of a Pipelined Data Transformation with Fanout

Diagram/Figure from Oracle Documentation
Data Warehouse: Physical Design: Extraction, Transformation & Loading (ETL)

- Error Handling:
  - Business Rule Violations.
  - Data Rule Violations.

- Filtering of Erroneous Data:
  - Identification
  - Separation

- Error Tables
Change Data Capture

- Prior Techniques:
  - Table Differencing (MINUS Operators).
  - Change-Value Selection.

- The new wave: Change Data Capture
  - Synchronous CDC (Trigger-based).
  - Asynchronous CDC (Log-based).
Change Data Capture: The Holy Grail of Data Warehousing

Near Real-Time Data Warehousing
(Trickle-Feed Data Acquisition & Delivery)

- Publish & Subscribe Model:
  - DBMS_CDC_PUBLISH
  - DBMS_CDC_SUBSCRIBE
- Advanced Queues
- Oracle Streams
Change Data Capture

Example of Publishers in a CDC System

Diagram/Figure from Oracle Documentation
Change Data Capture

Example of a Subscriber in a CDC System

Diagram/Figure from Oracle Documentation
Change Data Capture

Example of Synchronous CDC

Diagram/Figure from Oracle Documentation
Change Data Capture

Example of Asynchronous CDC

Diagram/Figure from Oracle Documentation
Change Data Capture

Example of Asynchronous Autolog Online CDC

Diagram/Figure from Oracle Documentation
Change Data Capture

Example of Asynchronous Distributed Hotlog CDC

Diagram/Figure from Oracle Documentation
Change Data Capture

Configuration of Streams Propagation used to achieve Asynchronous Distributed Hotlog CDC

Diagram/Figure from Oracle Documentation
Performance: Materialized Views & Query Rewrites

Steps involved in a Query Rewrite Process

Diagram/Figure from Oracle Documentation
Performance: Materialized Views & Query Rewrites

- Materialized Views are Pre-computed/Pre-generated tables made up of summaries/queries.

- **EXAMPLE:** CREATE MATERIALIZED VIEW join_sales_time_product_mv ENABLE QUERY REWRITE AS 
  SELECT p.prod_id, p.prod_name, t.time_id, t.week_ending_day, s.channel_id, s.promo_id, s.cust_id, s.amount_sold FROM sales s, products p, times t WHERE s.time_id=t.time_id AND s.prod_id = p.prod_id;
Performance: Materialized Views & Query Rewrites

- Query Rewrite Types:
  - Text Match Rewrite
  - Join Back
  - Aggregate Computability
  - Aggregate Rollup
  - Rollup Using a Dimension
  - Filtering the Data (Subset of Data)
  - PCT Rewrite
  - Multiple Materialized Views
Performance: Common Schema Models

- **Relational Schemas (OLTP Systems)**
  - 1\textsuperscript{st} Normal Form (1NF) to 6\textsuperscript{th} Normal Form (6NF)
  - 3NF is mainstream and widely used:
    - “Every non-prime attribute is non-transitively dependent on every key of the table”.
    - “The key, us the whole key, and nothing but the key”.

- **Dimensional Schemas (Data Warehouses):**
  - Star Schemas (Highly Recommended)
  - Snowflake Schemas
Performance: Common Schema Models

- OLTP Schema Model: 3rd Normal Form (3NF):

Example of a small 3NF Schema

Diagram/Figure from Oracle Documentation
Performance: Materialized Views & Query Rewrites

- Dimensional Schema Model: Star Schema:

  ![Example of a Star Schema](Diagram/Figure from Oracle Documentation)
Performance: Materialized Views & Query Rewrites

- Dimensional Schema Model: Snow-flake Schema:

Example of a Snow-flake Schema

Diagram/Figure from Oracle Documentation
Performance: Aggregation

- Cubes
- Rollups
- Grouping Functions
- Composite Columns
Performance: Aggregation

Example of Logical Cubes & Views

*Diagram/Figure from Oracle Documentation*
Tools: Oracle Data Integrator

Data Integration
Challenges & Emerging Solutions

- Increasing data volumes; decreasing batch windows
- Shift from E-T-L to E-LT
- Non-integrated integration
  - Convergence of integration solutions
- Complexity, manual effort of conventional ETL design
  - Shift from custom coding to declarative design
- Lack of knowledge capture
  - Shift to pattern-driven development

Oracle Data Integrator Overview
Diagram/Figure from Oracle Documentation
**Tools: Oracle Data Integrator**

**Oracle Data Integrator**
Data Movement and Transformation from Multiple Sources to Heterogeneous Targets

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Key Differentiated Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Heterogeneous “E-LT”</td>
</tr>
<tr>
<td>Flexibility</td>
<td>Active Integration Platform</td>
</tr>
<tr>
<td>Productivity</td>
<td>Declarative Design</td>
</tr>
<tr>
<td>Hot-Pluggability</td>
<td>Knowledge Modules</td>
</tr>
</tbody>
</table>

*Oracle Data Integrator Overview*
*Diagram/Figure from Oracle Documentation*
Tools: Oracle Data Integrator

E-LT Architecture
High Performance

Conventional ETL Architecture

Transform in Separate ETL Server
- Proprietary Engine
- Poor Performance
- High Costs
- IBM & Informatica’s approach

Next Generation Architecture

Transform in Existing RDBMS
- Leverage Resources
- Efficient
- High Performance

“E-LT”

Benefits
- Optimal Performance & Scalability
- Easier to Manage & Lower Cost

Oracle Data Integrator Overview

Diagram/Figure from Oracle Documentation
Tools: Oracle Data Integrator

Oracle Data Integrator Overview

Diagram/Figure from Oracle Documentation
Tools: Oracle Warehouse Builder

- Computer Aided Software Engineering (CASE) Tool
- Rich Oracle-centric Data Warehouse building feature-set
- Schema Modeling
- Mapping
- Data Quality
- Data Profiling
Tools: Oracle Warehouse Builder

- CASE Tool:: Practical Usage:
  - Design, Build & Deploy Schema Structures e.g. Tables, Indexes, External Tables etc.
  - Design, Build & Deploy Mappings
  - Design, Build & Deploy Transformations
  - Design, Build & Deploy Dimensions
  - Design, Build & Deploy Cubes
  - Design, Build & Deploy Queues
  - 11g Rel. 2 (Trickle-feed Data Acquisition & Delivery)
Tools: Comparison of Oracle Data Integrator (ODI) with Oracle Warehouse Builder (OWB)

- Both tools complement each other: However there is some overlapping functionality.

- Oracle’s product direction is to merge the tools into a single product bringing the best of both worlds into a single consolidated offering in the near future. For now, both tools are separately supported.
Project Planning: Approach: Bill Inmon vs. Ralph Kimball

- **Bill Inmon's Model: Top-Down Design:** “Data warehouse is one part of the overall business intelligence system. An enterprise has one data warehouse, and data marts source their information from the data warehouse. In the data warehouse, information is stored in 3rd normal form.”

- **Ralph Kimball's Model: Bottom-Up Design (Recommended):** “Data warehouse is the conglomerate of all data marts within the enterprise. Information is always stored in the dimensional model.”
Project Planning: Phased Implementation

- **Business Requirements:**
  - Scope / Users / Initiatives
  - Risk Assessment
  - **Key Requirements / Conceptual Design:**
    - Dimensional Modeling: Star Schemas
    - Reporting
    - OLAP
    - Data Mining
    - Data Marts

- **Technical Implementation:**
  - Future-proofing
  - Infrastructure: Hardware/Software
  - Human Resources
  - Technical Execution
Physical Implementation: Software Infrastructure: RDBMS: Its all about blazing performance!

- Recommendation: Oracle 11g Rel. 2 Enterprise Edition with OLAP, XML DB & OWB options
- Recommendation: >= 2-Node Real Application Cluster Configuration
- Recommendation: DataGuard with DataGuard Broker for Business Continuity / Automatic Failover
Physical Implementation: Software Infrastructure: RDBMS: Recommendations: Its all about blazing performance!

- >= 16KB Block Size
- QUERY_REWRITE_ENABLED = TRUE
- STAR_TRANSFORMATION_ENABLED = TRUE
Physical Implementation: Hardware Infrastructure: Network: Recommendations: Its all about blazing performance!

- Network Backbone Infrastructure: >= GigE (1 Gigabits per second)
- High-Speed SAN Infrastructure: >= 8Gb/s (8 Gigabits per second backbone): Fibre-Channel, Infini-band etc.
- High-Speed SAN: Drives: Fibre-Channel, Serial Attached SCSI (SAS) Drives etc.
- SAN Mirroring: RAID 1+0 (10) for PROD
- SAN Mirroring: RAID 5 for DEV, QA, TEST
Physical Implementation: Hardware Infrastructure: Servers: Recommendations: Its all about blazing performance!

- Clustered Low-Cost Commodity Hardware
- \( \geq \) Dual-Socket Quad-Core Processors with Hyper-Threading
- \( \geq \) 64GB RAM
- \( \geq \) 2.5Ghz Processor Clock Speeds
- Dual HBAs / HCAs \( \geq \) 8Gb/s
Summary

- To summarize, DataWarehousing in Oracle is proven, stable and fast and is used by entities, corporations and organizations worldwide to meet their BI/Warehousing needs. Learn more at Oracle's Datawarehousing Homepage.