A Brief Introduction to the Oracle Database Server

Dominik Lübbers
Dr. Christoph Quix
Outline

- Introduction and hints to further information
- Data View: What are the central components of an Oracle DB?
  - Basic schema objects: tables + constraints
  - Oracle in Action I: Travel Scenario in Oracle
  - Advanced schema objects and properties
    - Partitions, Clusters, Views, Indexes, Stored Procedures, Triggers
  - Oracle in Action II
  - Additional schema objects, Authentication + Data Dictionary
- Process View
  - How is this data processed?
  - Basic Components of an Oracle server
- Data Access and Application Interfaces
Presence on the DBMS Market

- One of the first commercial RDBMS (~25 years ago)
- One of the three DBMS giants [IDC 2004]
  - Oracle (39.8% market share RDBMS)
  - IBM DB2 (31.3% market share RDBMS)
  - Microsoft (12.1% market share RDBMS)
- Revenue in DBMS (2001): $13.6 billion
- Some reference customers of Oracle DBMS:
  - Amazon.com (10 TB data warehouse)
  - Southwest Airlines (High availability, scalability)
- "Post-relational" DBMS
Data View

What data is stored?
How is it grouped and organized?
**Oracle Schema**

- Container for logical structures of database
  - Tables
  - Views
  - Indexes
  - Stored procedures
  - Triggers
  - ...

- Do not mix up with relational (database) schema
- 1:1-relationship to users
Tables

- Implementation of "relation"

- Available data types (excerpt):
  - CHAR(n): fixed-length character strings (blank-padded)
  - VARCHAR2(n): variable-length character strings (max. length must be specified)
  - NCHAR(n), NVARCHAR(n): Unicode character strings
  - NUMBER(prec, scale)
  - DATE: Timestamps
  - LOBs: BLOB, CLOB, NCLOB…: LargeOBjects (max. 4 Gbyte)
  - XMLType
  - User-defined types...

- SQL-Example:
  
  ```sql
  CREATE TABLE Employee
  (ID NUMBER(10,0), Name VARCHAR(20))
  ```
Integrity Constraints

- Intrarelational constraints:
  - NOT NULL, UNIQUE, PRIMARY KEY, CHECK(<condition>)

- Referential integrity (foreign keys)
  - E.g.: ... G_AIRPORT_ID int REFERENCES G_Airport(ID)
  - Different ON DELETE reactions

- Deferring of constraints
  - Deferrable / not deferrable
  - Initially deferred / initially immediate

- Interrelational constraints
  - SQL-Standard: CREATE ASSERTION ...
  - Not available in ORACLE
  - Has to simulated by Triggers (see later)
Oracle in Action I

Definition of tables and constraints for the travel scenario

Sample queries against this database
Partitions

- Split table data according to a *partitioning attribute*
- Reduces index accesses

→ Shorter access time for queries with partitioning attribute
Clustered Tables

- Store related rows (clustered key) of different tables together
- Increases performance if tables are frequently joined

**Example**: Clustered Tables

<table>
<thead>
<tr>
<th>Employees</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>employee_id</td>
<td>last_name</td>
<td>department_id</td>
</tr>
<tr>
<td>201</td>
<td>Hartstein</td>
<td>20</td>
</tr>
<tr>
<td>202</td>
<td>Fay</td>
<td>20</td>
</tr>
<tr>
<td>203</td>
<td>Mavis</td>
<td>40</td>
</tr>
<tr>
<td>204</td>
<td>Baer</td>
<td>70</td>
</tr>
<tr>
<td>205</td>
<td>Higgins</td>
<td>110</td>
</tr>
<tr>
<td>206</td>
<td>Götz</td>
<td>110</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Departments</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>department_id</td>
<td>department_name</td>
<td>location_id</td>
</tr>
<tr>
<td>20</td>
<td>Marketing</td>
<td>1900</td>
</tr>
<tr>
<td>110</td>
<td>Accounting</td>
<td>1700</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clustered Key</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>department_id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>marketing</td>
<td>1800</td>
</tr>
<tr>
<td>201</td>
<td>Hartstein</td>
<td>...</td>
</tr>
<tr>
<td>202</td>
<td>Fay</td>
<td>...</td>
</tr>
<tr>
<td>205</td>
<td>Higgins</td>
<td>...</td>
</tr>
<tr>
<td>206</td>
<td>Götz</td>
<td>...</td>
</tr>
</tbody>
</table>
Views, Materialized Views

- Treat query result as a table
- Views can be used nearly whenever a table is used
- Views are **logical structures**, they do not consume any storage space (except for their definition)
- Advantages:
  - Additional level of data security
  - More convenient query formulation
  - Isolate applications from changes in definitions of base tables
- Snapshots of views: Materialized views (DW-applications)
Indexes

- Creates fast access path via indexed attributes
- Query optimizer decides for the fastest access path
- Unique vs. Nonunique Indexes
- Different Implementations:
  - B-tree
  - Bitmap index
  - Reverse key indexes
  - Function-based indexes
- Store data in index: Index-organized tables
Stored Procedures and Functions

- Sequence of PL/SQL statements with call interface (in/out variable declarations)
- Integration of operational aspects in DB is similar to ideas of object-oriented paradigm (with its benefits)
- Example:
  ```plsql
  Procedure hire_employee (last_name VARCHAR2, job_id VARCHAR2)
  BEGIN
      INSERT INTO employees VALUES (emp_sequence.NEXTVAL, last_name, job_id);
  END;
  ```
- Help enforcing data integrity and data security
- Functions can be used in SQL statements
Packages

- **Package**
  - Group of related PL/SQL procedures
  - Consists of:
    - Specification (declarations) and
    - Body (definitions)
  - Advantages
    - Encapsulation of logically related code => better organization
    - Private and public procedures, functions, variables, constants
Triggers

- Reaction to database events
- Trigger types
  - INSERT or UPDATE or DELETE
  - BEFORE or AFTER
  - ROW or STATEMENT
- Triggers are e.g. PL/SQL program units
- INSTEAD OF trigger replaces standard table operation
- Can solve complex data integrity rules if declarative integrity constraints are not sufficient
Oracle in Action II

Definition of views, indexes and triggers for the travel scenario
Sequences

- Provide sequential series of numbers:
  - `seq_name.nextval`
- Simplifies generation of unique identifiers

Synonyms

- Alias for schema object (e.g. in different schema)
- Especially useful in distributed database environments (location transparency)
- Public vs. private synonyms
Users and Privileges

- User authentication
  - By operating system
  - By network
  - By oracle database
- User owns a schema
- User can be granted object and system privileges
- SQL examples:
  - GRANT CREATE SESSION, CREATE TABLE TO user1;
  - GRANT EXECUTE ON dbms_pipe TO public;
  - GRANT UPDATE(ename, sal) ON emp TO user1 WITH GRANT OPTION
  - REVOKE DELETE ON emp FROM scott
Roles and Profiles

- **Roles**
  - Named sets of system and schema object privileges
  - Comparable to user groups in OS
  - Roles can inherit privileges from other roles
  - Assign users to roles

- **Profiles**
  - Named sets of resource limits, e.g. for:
    - CPU time
    - I/O operations
  - Prevent uncontrolled consumption of valuable resources
  - Are assigned to users
Storage Structures - Overview

Logical:
- Database
  - Tablespace
    - Segment
      - Extent
        - Oracle block

Physical:
- Data file
  - O/S Block
Data Dictionary I

- Collection of read-only tables describing the database structure
- Contains information about:
  - Logical DB structure (schema objects, tablespaces, …)
  - Physical DB structure (data files, …)
  - Users, Roles and Profiles
  - Statistics for use by the query optimizer
  - …
- Automatically updated by server processes when database structure changes
**Data Dictionary II**

- Table content and table structure can be queried by the same interface!

- Prefixes for views on Data Dictionary tables:
  - **USER**: user's view (what is in the user's schema)
  - **ALL**: expanded view (what the user can access)
  - **DBA**: Database administrator's view (what is in all users' schemas)

- Some examples:
  - `SELECT object_name, object_type FROM USER_OBJECTS;`
  - `SELECT owner, table_name, tablespace_name FROM ALL_TABLES;`
Process View

How are data requests processed?

What are the main internal components of an Oracle server?
Client / Server Architecture

Client

Server

Oracle server

Server process

User

User process

Client

User
Oracle Instance

- Host
  - runs
  - manages

Instance

Database

Background processes
Server Components: SystemGlobalArea (SGA)

- Database buffer cache
- Redo log buffer
- Shared pool
  - Library cache
    - Statement text
    - Parsed code
    - Execution plan
  - Data dictionary cache
Background Processes and PGA

**Background processes**
- DBW: Database writer process
  - Write modified (dirty blocks) to disk
- LGWR: Log writer process
  - Write log entries to disk

**Program Global Area (PGA)**
- One per server process
- Sort area
- Session information
- Stack space
Summary - Process View
Data Access and Application Interfaces

How can a user access the database?
How can application programs access the DB?
Data Access: SQL Statements

- **DML: Data Manipulation Language**
  - SELECT, UPDATE, INSERT, DELETE, ...

- **DDL: Data Definition Language**
  - CREATE, ALTER, DROP, GRANT, REVOKE, ...

- **Transaction control statements**
  - COMMIT, ROLLBACK, SAVEPOINT, ...

- **Session control statements**
  - ALTER SESSION, SET ROLE, ...

- **System control statements**
  - ALTER SYSTEM
<table>
<thead>
<tr>
<th><strong>Data Access: PL/SQL</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedural programming language with:</td>
</tr>
<tr>
<td>- Embedded SQL expressions</td>
</tr>
<tr>
<td>- Variables, Constants</td>
</tr>
<tr>
<td>- Exceptions</td>
</tr>
<tr>
<td>- Cursors (record-oriented processing of Oracle data)</td>
</tr>
</tbody>
</table>

Data types suitable for database programming:
- NULL VALUES
- TABLE datatype

PL/SQL blocks can be stored in the database and e.g. used as triggers.
Application Programs and Databases

Three approaches to database programming:
- Invent a new language (like PL/SQL)
- Embed database commands in a general-purpose language
  - Precompiler translates database commands to library function calls
  - SQL-statements are parsed and checked at compile time
  - Examples: SQLJ, Pro*C, ...
- Provide library of database functions
  - Formulation of SQL-queries at runtime
  - Examples: SQL/CLI, JDBC, ODBC, ...
**API-based Data Access: JDBC**

- **Dynamic SQL via JDBC-driver at the client**

```java
// Load Oracle driver
Class.forName("oracle.jdbc.OracleDriver");
// Connect to the local database
Connection conn = DriverManager.getConnection(
    "jdbc:oracle:thin:@robinie:1521:TESTBASE","edb_xxx","password");
// Query the DRINKS table
Statement stmt = conn.createStatement();
ResultSet rset = stmt.executeQuery("SELECT * from DRINKS");
// Print out the first attribute
while (rset.next ())
    System.out.println (rset.getString (1));
// close the result set, statement, and the connection
rset.close();
stmt.close();
conn.close();
```

- **Oracle 10g allows storage and execution of Java code in database**
Some Oracle Tools

- Execute PL/SQL scripts (including SQL statements)
  - SQL*Plus
    - Graphic (Java-based) front end
    - Command line
    - Web interface

- Administration of database
  - Oracle Enterprise Management Console
    - Java-based front end
    - Web interface (recommended for Oracle 10g)

- Application Development
  - Oracle JDeveloper, Oracle Designer, ...
What happens next...

- Evaluate it on your own
  - Create simple schema objects using our Oracle server on ROBINIE
    - Access tickets available at our office
  - Full enterprise server available for evaluation purposes
    - Register and download from Oracle Technology Network www.otn.oracle.com
    - All standard platforms
  - Extensive documentation
    - Good starting point:
      - Oracle 2day DBA
      - Database System Concepts