

Table of Contents

Slide 1: Development of Oracle applications - Best practices.....	1
Development of Oracle applications - Best practices.....	1
Slide 2: Documentation.....	1
Slide 3: Keys, constraints and indices.....	1
Primary key:.....	1
Foreign key:.....	1
Slide 4: Keys, constraints and indices (2).....	1
Primary and foreign keys syntax examples:.....	2
Unique keys.....	2
Other constraints.....	2
Slide 5: Keys, constraints and indices (3).....	2
Indices:.....	2
Types of indices:.....	2
Slide 6: Bind Variables.....	3
Place-holder variables in SQL statements that are substituted with concrete values before statement execution e.g:.....	3
Bind variables help:.....	3
Practically all database APIs support bind variables however naming and syntax conventions are different.....	3
Slide 7: Full table scans.....	3
Slide 8: Connection management.....	3
Opening a connection to a database is very expensive from Oracle point of view:.....	3
Client-server applications:.....	4
N-Tier applications:.....	4
Slide 9: Summary	

Development of Oracle applications - Best practices

Jacek Wojcieszuk, IT/PSS March, 2006

Slide 2: Documentation

Oracle software comes together with extensive documentation. In case of RDBMS 10gR2 there are ~50k pages of documentation. Its well structured: there are books for both DBAs and developers. Some books are strictly technical and some give an overview. It is the best source of information for application developers. However the attempt to read it entirely is the best way to fail the project.

Most useful pieces:

- Application Developer's Guide - Fundamentals
- SQL Reference
- PL/SQL User's Guide and Reference
- Concepts

Oracle documentation at CERN:

- oradoc.cern.ch
- Documentation sets for different products and versions
- Use documentation for product version you use!

Slide 3: Keys, constraints and indices

Primary key:

- Column or set of columns that uniquely identifies table rows
- Every table should have a primary key defined
- Primary key protects from entering duplicated rows to a table
- Oracle implicitly creates a unique index on primary key columns

Foreign key:

- Relation (e.g. master-detail) between 2 tables should be formalized by creating a foreign key constraint
- Foreign key helps to keep data integrity and facilitates cascade operations
- Oracle DOES NOT create implicitly indices on foreign key columns so if related tables are supposed to be joined one needs to create such index by hand.

Slide 4: Keys, constraints and indices (2)

Primary and foreign keys syntax examples:

```
ALTER TABLE employees ADD CONSTRAINT employees_pk
PRIMARY KEY (id);
```

```
ALTER TABLE departaments ADD CONSTRAINT employees_pk
    PRIMARY KEY (dept_id);
ALTER TABLE employees ADD CONSTRAINT departaments_fk
    FOREIGN KEY (dept_id) REFERENCES departaments (dept_id);
```

Unique keys

- Column or set of columns other than primary key that uniquely identifies rows in the table
- Oracle implicitly creates index on unique keys
- In contrary to primary keys, unique keys allow NULL values

```
ALTER TABLE employees ADD CONSTRAINT employees_unq UNIQUE (lname);
```

Other constraints

Usually it is better to enforce data integrity constraints on database server side than on client/middle tier side:

- Central management
- Easier maintenance
- Better performance
- Allowed constraints:
 - ◆ NOT NULL
 - ◆ CHECK

```
ALTER TABLE employees ADD CONSTRAINT salary_chk
    CHECK (salary BETWEEN 1000 AND 50000);
```

Slide 5: Keys, constraints and indices (3)

Indices:

- They are like indices in books -> they speed up lookups
- BUT: insert into a table with 3 indices is ~10 times slower than into table without indices
- It is important to find for each table a minimal set of indices
- It is usually better to not create indexed on small tables (< 50 rows)

```
CREATE INDEX emp_dept_id_idx ON employees (dept_id);
```

Types of indices:

- Unique/non-unique
- B-Tree
 - ◆ Best for high cardinality columns
- Bitmap indices
 - ◆ Best for low cardinality columns
 - ◆ BUT: have very negative impact on performance of DML operations and their scalability

Slide 6: Bind Variables

Place-holder variables in SQL statements that are substituted with concrete values before statement execution e.g:

```
SELECT "schedulerJobId" FROM CMS_DASHBOARD.JOB WHERE "jobId"= 2001;  
SELECT "schedulerJobId" FROM CMS_DASHBOARD.JOB WHERE "jobId" = :job_id;
```

Bind variables help:

- To reduce CPU consumption (less hard parses)
 - ◆ 5000 executions of query above take either 2.4 or 18.1 s depending on whether bind variable has been used or not.
- To improve application scalability (less Oracle internal locking)
- To decrease DBAs anger
- To improve code security
 - ◆ Bind variables protect from SQL injections.
- To make code more readable and easier to maintain
 - ◆ Concatenated strings with hard-coded values are usually less readable than strings with bind variables

Practically all database APIs support bind variables however naming and syntax conventions are different

- ◆ If given API does not support them don't use it!

Slide 7: Full table scans

- Happen when:
 - ◆ a SQL statement does not contain the WHERE clause
 - ◆ The WHERE clause is not very selective
 - ◆ Any of the columns used in the WHERE clause predicate does not have an index.
- Cause excessive IO
- Slow down the whole system
 - ◆ IO subsystem is a usual bottleneck of the database system
- Are acceptable only for small tables!

Slide 8: Connection management

Opening a connection to a database is very expensive from Oracle point of view:

- New OS process has to be forked
- Session context has to be created inside Oracle instance
- User has to be authenticated and authorized

Client-server applications:

- Database session should be created at the beginning of the client program execution and reused during its lifetime
- EXCEPTION: if client is idle for a long period of time it is better to close the session and free server-side resources

Place-holder variables in SQL statements that are substituted with concrete values before statement execution

N-Tier applications:

- Connection pooling should be used in the middle tier
- Most of the API support connection pooling

Slide 9: Summary

- Most useful tips for Oracle application developers can be found on Wiki:
 - ◆ <https://uimon.cern.ch/twiki/bin/view/PSSGroup/GeneralAdvices>
- Additional tips one can find on Oracle pages:
 - ◆ <http://otn.oracle.com>
 - ◆ <http://asktom.oracle.com>
- For some advice it is always possible to ask us:
 - ◆ physics-database.support@cern.ch
- We continuously improve our monitoring facilities so:
 - ◆ Be careful Big Brother is watching

This topic: PSSGroup > ReferenceSlideshow
Topic revision: r1 - 2006-03-29 - unknown



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback