Jim Schmidt

March 30, 2009

# 1 todo

discuss key types UUID, sequences,

discuss reverse engineering

creating and destroying your data for testing

transaction management

getting an Oracle Connection

using ejb3 notation

javadoc

describe the differences between an oracle transaction and a hibernate transaction

todo show what happens in a conflict between two updates of the same object in a level one

replace the datasource in the application-context

describe the application-context

DAO and DTO
todo describe what a DAO and a DTO are

Installing

# 2 Download the Source

# 3 Create the Database Objects

## 3.1 Create The Schema

The most succinct way to describe the user and privileges required is to show the create user script.

```
drop user &&username cascade;
create user &&username identified by &&password;
alter user &&username default tablespace users;
alter user &&username quota unlimited on users;
grant create table    to &&username;
grant create session  to &&username;
grant create sequence to &&username;
grant create view     to &&username;
grant alter session   to &&username;
```

## 3.2 Create the database Objects

See 8.1 (page 4)

# 4 Configure Your Connections

todo edit etc/application-context

# 5 Reverse Engineer into Hibernate

todo describe the Revenge class

## 5.1 Create mapping files

## 5.2 Create POJOS

# 6 Populate the data

package com.dbexperts.sales.populate.DataPopulator

# 7 Run the tests

# 8 The Scripts

## 8.1 Creating the User

```
drop user &&username cascade;
create user &&username identified by &&password;
alter user &&username default tablespace users;
alter user &&username quota unlimited on users;
grant create table    to &&username;
grant create session  to &&username;
grant create sequence to &&username;
grant create view     to &&username;
grant alter session   to &&username;
```

## 8.2 DDL

```
spool sale
set echo on

create table product_etl
(
product_etl_id    number(9) not null,
upc10          varchar2(10),
product_status varchar2(1),
descr          varchar2(50),
narrative      clob
);


create table product
(
product_id    number(9) not null,
upc10          varchar2(10) not null,
product_status varchar2(1) not null,
descr          varchar2(50),
narrative      clob
);

comment on table product is
'Product Master

';
```

```
comment on column product.upc10 is '10 digit Universal Product Code, the American equiva

comment on column product.status is
'The status of the item.
 A - Active
 S - Setup
 I - Inactive

';

alter table product add constraint product_pk
primary key (product_id);

alter table product add constraint product_uq unique (upc10);

create sequence product_seq;
--
--
--

create sequence customer_seq;

-- todo everything should be nullable until setup
-- having inside_salesperson_id not null allows for a great deal of query optimization
-- therefor this should not be dependent on customer status
-- todo should have a table to hold until actually set up
create table customer
(
customer_id number(9) not null,
customer_status varchar2(1),
name         varchar2(30),
addr_1       varchar2(30),
addr_2       varchar2(30),
city         varchar2(25),
state        varchar2(2),
zip_cd       varchar2(10),
outside_salesperson_id number(9),
inside_salesperson_id  number(9)  not null
);

-- todo comment every column and what the sequence is
alter table customer add  constraint customer_pk primary key
(customer_id);
```

```
create bitmap index customer_ak1 on customer(outside_salesperson_id);

create bitmap index customer_ak2 on customer(inside_salesperson_id);

comment on column product.upc10 is
' Why is this varchar2 if it is a number?';

create table sale
(
sale_id    number(18) not null,
ship_dt     date not null,
qty         number(13,5) not null,
        product_id number(9) not null,
customer_id number(9) not null
);

create sequence sale_seq;

alter table sale add constraint sales_pk primary key (sale_id);

alter table sale
add constraint s_c_fk
foreign key (customer_id)
references customer(customer_id);

alter table sale
add constraint s_p_fk
foreign key (product_id)
references product(product_id);

alter table sale add constraint sale_pk primary key (sale_id);

--
--
--
create table salesperson
(
salesperson_id number(9),
display_name   varchar2(40),
first_name     varchar2(16),
last_name      varchar2(20)
);

alter table salesperson add constraint salesperson_pk
primary key (salesperson_id);
```

```
create sequence salesperson_seq;
/*
 alter table customer
 add constraint c_is_fk
 foreign key (inside_salesperson_id)
 references salesperson(salesperson_id);
*/

-- todo restore after figuring out how to get reveng to show reference also

-- alter table customer add constraint c_sp_fk1  foreign key
-- (inside_salesperson_id) references salesperson (salesperson_id);


-- alter table customer add constraint c_sp_fk2  foreign key
-- (outside_salesperson_id) references salesperson (salesperson_id);

create view customer_sale_product
as
select
c.name,
c.addr_1,
c.addr_2,
c.city,
c.state,
c.zip_cd,
p.upc10,
p.product_status,
p.descr         product_descr,
p.narrative,
sp.salesperson_id inside_salesperson_id,
sp.display_name inside_rep__display_name,
sp.first_name   inside_rep_first_name,
sp.last_name    inside_rep_last_name,
s.sale_id,
s.ship_dt,
s.qty,
        s.product_id,
s.customer_id
from
customer c,
product p,
salesperson sp,
  sale s
where   c.inside_salesperson_id = sp.salesperson_id(+) and
s.product_id = p.product_id and
```

```
        c.customer_id = s.customer_id;

alter view  customer_sale_product
add constraint customer_sale_product_pk
primary key (sale_id) disable novalidate ;

create global temporary table gtt_number
(
nbr number
) on commit delete rows;
```

# 9 Todo

- Need to get all of the events, waits and sql from v$session to see what is truly going on.
- Get correct Sequence generation Strategy
- Get the ant file to use the hbm.xml when generating the pojos, not doing it directly
- Show how to use the ejb style
- show how to get a native oracle connection and process arrays for fetching
- show how to customize the OracleDatasource and how it can be tuned
- show how to generate and view Javadoc in eclipse
- create a generate javadoc task
- exclude from cvs
- get column comments in the javadoc
- after you add a foreign key a reference to the object
- show various ways to implement persistence including delimited files
- explain the difference between an interface and a package specification, multiple implementations

Note that the default behavior of Object representation of foreign keys are to objects of the type being referenced, which can be goofy.
Reverse Engineering,
Get the correct sequence generator in place
In or

## 9.1 Install

Create a working environment

- install the Operating System (centos / kubuntu / ??????)
- install Oracle RDBMS
- install Oracle EM
- install Apex
- Install Eclipse
- Install SQLdeveloper
- Install Apache
- Install Tomcat

# 10 Unit tests

**Abstract**

Hibernate is a widely used Object Relational Mapping tool, used to facilitate interaction with a SQL database from Java.

During the 2009 Hotsos Oracle Performance meeting it was widely accepted that there are many performance problems associated with Hibernate but that "nothing could be done about it".

This article intends to dispel that myth with a brief description of the configuration of Hibernate and some small working examples that demonstrate the points made in this article. This is not a tutorial on Java or Hibernate. It is expected that these working examples will function as a prototype for in house use when warranted.

A 99 percent improvement performance on something that burns a few CPU seconds a day and takes weeks to get into production is hardly cost effective. Frequently run queries that burn large amounts of computer resources and create contention for Oracle serialized (latched, locked, enqueued, pinned...) resources are candidates for modification. . Just because something is sub-optimal does not mean that it should be improved. The cost of "implementing" an improvement must be less than the value of the benefit derived from the improvement.

# 11 todo

Why Datasources.xml

# 12 Introduction

I intend to show various ways to improve Oracle performance when using Hibernate. The java developers may not like all of these techniques. Hibernate does a good job when used as intended... (todo elaborate)

It's payback time.

DBA: Your execution plan is seriously suboptimal due to latch contention on busy block buffers due to consistent read on the application control table accross all of these sessions.

Java: I could read and cache the application control table entries and associate them with a HashMap to my detail entries and write a strategy to control my secondary cache in attempt to keep in sync with committed transactions but the change notification mechanism doesn't ship until Oracle 11 which sure isn't ready for production use yet. Seems like you're putting the work of database management on the app side and we are re-inventing the wheel. Now I could create a AOP cut point to send a JMS ....

todo transaction demarcation

### 12.1 Terminology

Child Relation - Frequently referred to as a detail table, this the “many end” of a “one to many” relationship.

### 12.2 Hibernate Strengths

todo flesh out

### 12.3 Hibernate Weakness

To properly

1. Why you should create constraints on views if you are using Hibernate Reverse Engineering.
2. Describe the various fetch strategies and how they can be modified
3. Describe a mechanism to reduce hard parses.
4. Key generation
5. Using Views with Relations

### 12.4 Improving Performance

With a Session Managed Bean in hand a traversal to a *child relation* Hibernate may find it appropriate to fetch all of the todo associated with the other *Parent relation* records.

### 12.5 More Tips

## 13 Fetching Policies

### 13.1 Eager Fetch

An eager fetch assumes that the child tables of a given table will be visited and so will be fetched by Hibernate in advance of any attempt to reference the child relation.

### 13.2 Lazy Fetch

## 14 Declarative Fetch Policy

- Eager fetching.

# 15 Lets Get On With Tutorial

## 15.1 Create The Schema

I assume that the dba creates the schema and that the java developers reverse engineer the schema into hibernate.

Let's assume a schema 15.4 on 12

## 15.2 Create a Reverse Engineering Strategy

## 15.3 Generate the mapping files

## 15.4

Now Fetch the masters, create an ID list and fetch the details followed by a mapped association of details to master.