# Instructions

- Create account in https://apex.oracle.com/en/

- Request Workspace

- Run the following to create the data model

```sql

create table item(

    item varchar2(25) not null,

    dept number(4) not null,

    item\_desc varchar2(25) not null

);

create table loc(

    loc number(10) not null,

    loc\_desc varchar2(25) not null

);

create table item\_loc\_soh(

item varchar2(25) not null,

loc number(10) not null,

dept number(4) not null,

unit\_cost number(20,4) not null,

stock\_on\_hand number(12,4) not null

);

--- in average this will take 1s to be executed

insert into item(item,dept,item\_desc)

select level, round(DBMS\_RANDOM.value(1,100)), translate(dbms\_random.string('a', 20), 'abcXYZ', level) from dual connect by level <= 10000;

--- in average this will take 1s to be executed

insert into loc(loc,loc\_desc)

select level+100, translate(dbms\_random.string('a', 20), 'abcXYZ', level) from dual connect by level <= 1000;

-- in average this will take less than 120s to be executed

insert into item\_loc\_soh (item, loc, dept, unit\_cost, stock\_on\_hand)

select item, loc, dept, (DBMS\_RANDOM.value(5000,50000)), round(DBMS\_RANDOM.value(1000,100000))

from item, loc;

commit;

NOTE: While inserting records in the table item\_loc\_soh a message about tablespace quota exceeded appeared.

I have tried to run:

alter user wksp\_ruispratley quota unlimited on apex\_bigfile\_instance\_tbs2;

And

alter user wksp\_ruispratley quota 50M on apex\_bigfile\_instance\_tbs2;

And both failed.

To be able to continue the exercise I have changed the number of records in the tables:

ITEM 1000 records

LOC 100 records

If I had permissions, I would have increased user quota on tablespace and used the supplied number of records.

```

- in the Apex App Builder import the application StockApplication.sql. When opening the application for login will be the same email and password as registered at apex.oracle.com.

**\*\*NOTE: If you fail to complete any challenge please still reply with what was the thinking and why you were not able to complete.\*\***

[INFO]

> After finishing the problems, create a public repository in Github, push your code and send us the Github Public URL.

> If for some reason this is not possible, send us a zip folder containing a install script with all the required solution identified.

> The source contains a StockApplication.zip Apex application that can be deployed to better understand the challenge from user perspective.

**# Context**

Item loc stock an hand represents a snapshot table of stock in a specific moment for all items in all stores/warehouses for a retailer. In scenario where you have an Apex application that enables a view of stock per store/warehouse please consider the following:

 - this application has an very high user concurrency access during the entire day

 - the access to the application data is per store/warehouse

 - one of the attributes that most store/warehouse users search is by dept

**# Challenge**

**## Must Have**

**### Data Model**

Please consider that your reply for each point should include an explanation and corresponding sql code

**1. Primary key definition and any other constraint or index suggestion**

The primary keys should be created for unique values that identify one record

alter table ITEM add constraint ITEM\_PK PRIMARY KEY (ITEM, DEPT);

--I don’t see any mention to tablespace otherwise would be in indexes tablespace

# alter table LOC add constraint LOC\_PK PRIMARY KEY (LOC);

alter table ITEM\_LOC\_SOH add constraint ITEM\_LOC\_SOH\_PK PRIMARY KEY (ITEM, LOC, DEPT);

Foreign keys to enforce integrity between data

alter table ITEM\_LOC\_SOH add constraint ILS\_IT\_FK (ITEM, DEPT) references ITEM (ITEM, DEPT);

alter table ITEM\_LOC\_SOH add constraint ILS\_LOC\_FK (LOC) references LOC (LOC);

Also indexes for the foreign keys to avoid contention on delete operations on parent tables for instance

create index ILS\_IT\_FK\_IDX on ITEM\_LOC\_SOH (ITEM, DEPT);

create index ILS\_LOC\_FK\_IDX on ITEM\_LOC\_SOH (LOC);

**2. Your suggestion for table data management and data access considering the application usage, for example, partition...**

With the primary keys and indexes created, the application behavior should be acceptable

Always try to use filters in tables for columns that are indexed

If the number of stores was small (like 10 for example) and a huge number of items and dept per store, the table could be partitioned by store but with 1000 stores having 1000 partitions in the table and creating a new one for each store does not seem correct

**3. Your suggestion to avoid row contention at table level parameter because of high level of concurrency**

Minimize number of DML operations not committed

For large DML operations try to use commands like FORALL plus the DML operation

If the row contention still exists analyze V$SESSION and V$ACTIVE\_SESSION\_HISTORY tables to see the behavior at runtime and if any query is a bottleneck analyze it and see actions to improve it by changing the query or adding any missing constraint, among others

**4. Create a view that can be used at screen level to show only the required fields**

CREATE OR REPLACE FORCE VIEW V\_LOC\_IT (ITEM, LOC, DEPT, UNIT\_COST, STOCK\_ON\_HAND)

as SELECT ILS.ITEM, ILS.LOC, ILS.DEPT, ILS.UNIT\_COST, ILS.STOCK\_ON\_HAND

FROM ITEM\_LOC\_SOH ILS;

**5. Create a new table that associates user to existing dept(s)**

First create the USERS table:

create table USERS(

    user varchar2(25) not null,

    user\_desc varchar2(25) not null,

constraint USERS\_PK PRIMARY KEY (USER)

);

Then create the DEPT table:

create table DEPTS(

    DEPT number(24) not null,

    DEPT\_DESC varchar2(25) not null,

constraint DEPTS\_PK PRIMARY KEY (DEPT)

);

Then create the association table

create table USERS\_DEPTS(

    USER varchar2(25) not null,

    DEPT number(24) not null,

constraint USERS\_DEPTS\_PK PRIMARY KEY (USER, DEPT)

);

Also as above the FKs and IDX should be created

**### PLSQL Development**

**6. Create a package with procedure or function that can be invoked by store or all stores to save the item\_loc\_soh to a new table that will contain the same information plus the stock value per item/loc (unit\_cost\*stock\_on\_hand)**

create table item\_loc\_soh\_cost(

item varchar2(25) not null,

loc number(10) not null,

dept number(4) not null,

unit\_cost number(20,4) not null,

stock\_on\_hand number(12,4) not null,

total\_cost number(24,4) not null,

constraint ITEM\_LOC\_SOH\_COST\_PK PRIMARY KEY (ITEM, LOC, DEPT)

);

CREATE PACKAGE pk\_store\_info AS

PROCEDURE migrate\_info (loc\_id NUMBER);

END pk\_store\_info;

/

CREATE PACKAGE BODY pk\_store\_info AS

PROCEDURE migrate\_info (loc\_id NUMBER) IS

TYPE myarray IS TABLE OF VARCHAR2(30);

l\_data myarray;

cursor c\_data\_loc\_id is

select rowid

from item\_loc\_soh

where loc = loc\_id

and loc\_id is not null

union all

select rowid

from item\_loc\_soh

and loc\_id is null;

BEGIN

begin

open c\_data\_loc\_id;

loop

fetch c\_data\_loc\_id BULK COLLECT INTO l\_data limit 50000;

forall i in 1 .. l\_data.count

insert into item\_loc\_soh\_cost (item, loc, dept, unit\_cost, stock\_on\_hand, total\_cost)

select item, loc, dept, unit\_cost, stock\_on\_hand, unit\_cost \* stock\_on\_hand

from item\_loc\_soh

where rowid = l\_data(i);

exit when c\_data\_loc\_id%NOTFOUND;

end loop;

commit;

close c\_data\_loc\_id;

end;

EXCEPTION WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: SQLCODE=' || SQLCODE || ' SQLERRM=' || SQLERRM);

END;

END pk\_store\_info;

/

**7. Create a data filter mechanism that can be used at screen level to filter out the data that user can see accordingly to dept association (created previously)**

select \*

from item\_loc\_soh ils

join user\_dept ud

on ud.dept = ils.dept

where ud.user = :user;

--:user – Bind variable to filter the username

**8. Create a pipeline function to be used in the location list of values (drop down)**

CREATE OR REPLACE FUNCTION location\_list

RETURN location\_list\_t PIPELINED AUTHID DEFINER IS

cursor c\_loc is

select loc

from loc;

BEGIN

for r\_loc in c\_loc

loop

PIPE ROW (r\_loc.loc);

end loop;

RETURN;

END;

/

SELECT COLUMN\_VALUE FROM TABLE (location\_list ())

**## Should Have**

**### Performance**

**9. Looking into the following explain plan what should be your recommendation and implementation to improve the existing data model. Please share your solution in sql and the corresponding explain plan of that solution. Please take in consideration the way that user will use the app.**

```sql

 Plan Hash Value  : 1697218418

------------------------------------------------------------------------------

| Id  | Operation           | Name         | Rows | Bytes | Cost  | Time       |

------------------------------------------------------------------------------

|   0 | SELECT STATEMENT    |              | 100019 | 40760 | 10840 | 00:00:03 |

| \* 1 |   TABLE ACCESS FULL | ITEM\_LOC\_SOH | 100019 | 40760 | 10840 | 00:00:03 |

------------------------------------------------------------------------------

Predicate Information (identified by operation id):

------------------------------------------

\* 1 - filter("LOC"=652 AND "DEPT"=68)

Notes

-----

- Dynamic sampling used for this statement ( level = 2 )

```

Having the columns indexed the access should be by INDEX and not TABLE FULL ACCESS

Running the explain plan for the same query:

explain plan for select \* from ITEM\_LOC\_SOH where loc = 652 and dept = 68;

select \* from table(dbms\_xplan.display);

Result:

**10. Run the previous method that was created on 6. for all the stores from item\_loc\_soh to the history table. The entire migration should not take more than 10s to run (don't use parallel hint to solve it :))**

Test - pk\_store\_info.migrate\_info;

**11. Please have a look into the AWR report (AWR.html) in attachment and let us know what is the problem that the AWR is highlighting and potential solution.**

The first thing that is noticeable in the AWR is the high number of logical reads

Most of the reads are from 2 JOBS (DBMS\_SCHEDULER) and for automatic indexing SYS\_AI\_MODULE

Assuming that the storage access is not the bottleneck than the information access is (queries)

# Create a new AWR after the indexes creation and monitor again to see if the values are the same or improved.

**## Nice to have**

**### Performance**

**11. Create a program (plsql and/or java, or any other language) that can extract to a flat file (csv), 1 file per location: the item, department, unit cost, stock on hand quantity and stock value.**

**Creating the 1000 files should take less than 30s.**

This script assumes that there is an oracle directory called my directory and that the oracle user calling it has access to it. In addition, it needs to have space to store all files.

create or replace procedure EXPORT\_INFO is

new\_file UTL\_FILE.FILE\_TYPE;

cursor c\_loc is

select DISTINCT ILSC.LOC

from item\_loc\_soh\_cost ILSC;

cursor c\_info (LOC\_ID) is

select ILSC.ITEM,

ILSC.DEPT,

ILSC.UNIT\_COST,

ILSC.STOCK\_ON\_HAND,

ILSC.TOTAL\_COST

from item\_loc\_soh\_cost ILSC

where ILSC.LOC = LOC\_ID;

begin

for r\_loc in c\_loc

loop

begin

new\_file:= UTL\_FILE.FOPEN('my\_directory', 'FILE\_'||r\_loc.loc, 'w');

for r\_info in c\_info (r\_loc.loc)

loop

UTL\_FILE.PUT\_LINE(new\_file, r\_info.ITEM||','||to\_char(r\_info.DEPT)||','||to\_Cahr(r\_info.UNIT\_COST)||','||to\_char(r\_info.STOCK\_ON\_HAND)||','||to\_char(r\_info.TOTAL\_COST));

end loop;

UTL\_FILE.FCLOSE(new\_file);

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Exception writing file ||'FILE\_'||r\_loc.loc||: SQLCODE=' || SQLCODE || ' SQLERRM=' || SQLERRM);

UTL\_FILE.FCLOSE(new\_file);

end;

end loop;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Exception: SQLCODE=' || SQLCODE || ' SQLERRM=' || SQLERRM);

RAISE;

end;





