TRAVEL EXPERT AGENCY

Full Document

SEPT 2019 - MAY 2020

SALLY JANG

Contents

Requirements Document

Interview Questions and Answers

Day 1 Questions and Answers

1. Hardware requirements? Budget?

   Want us to recommend, and will likely need new hardware for everything. Part of the budget of $500 000.

2. Network infrastructure currently available?

   None, except for internet and printer. Will need recommendations.

3. Point of contact?
   Greg.
4. Are you okay with staff requiring new training? How long do you need for training?
   Yes, just schedule it into the project schedule
5. What is it that you wish for the database (and later website) to do?

   Keep the data safe in a central location, in a consistent manner, because currently there is no way of looking up data.

6. Any issues with your current system?

   There is no current system.

7. When is the deadline?

   May 1st

8. Audit requirements?

   Answered in later meetings.

9. Any particular access/permissions you wish to give to different group/people?

   Answered in later meetings.

10. How long do you wish to keep data?

    At least 6 years, but as long as possible. Supplier supplies new data every 6 months.

11. Backup requirements? How frequently? How many copies?

    At least 2 different copies. They want recommendations.

12. Potential changes in business rules?

    Hope to streamline commissions.

13. What kind of deliverables would you like?

Answered in later meetings.

14. Did you want to do promotional events based off of the birthdays of the customers?

    They themselves cannot create any deals or offers. The suppliers are the ones that create the deals. However, they would like the ability to flag/notify different customers based upon the offers that their supplies gives them.

    Examples of things to keep track of are birthdays, anniversary days, food allergies, preferences like window seats, etc.

15. How does the commission system work? How do you want the commission verification system to work?

    Currently the system doesn't work. Sometimes the customer books, but cancels. The suppliers are supposed to send the commission fee within 60 days after the customer's trips end. They never know for sure if the supplier paid, customer didn't go on the trip, or if the payment is late.

16. Do you want some sort of system for booking fees?

    The booking fees are at the discretion of the agent. It's usually based on if the trip took a lot of research to put together or not. It's usually $25, but might be more.

17. What forms do you use/generate?

    Invoices and itinerary. They want them consistent as currently it's all done by hand and individually.

18. Would you like to store additional employee information? How do you deal with ambiguity in the database regarding employees who share initials? Would you like a formatted employee number?

    Flat the inconsistent data for customers. Flag the employees with same initials. Generate employee numbers.

19. Are there any preference for format of information in 'description' column of the sales table?

    They want it consistent, so make some sort of system.

20. What to do if no agents are referred on the sales table?

    Flag the data.

21. Any ideas about how to give promotional offers your clients would like?

    The offers are given by the suppliers and they don't make it.

22. What does the Product Supplier ID reference?

    Reference the table with the product.

23. …Questions about the inconsistent data found.

    For any inconsistent data, just flag them. More information will be given later if needed. Data may be cleaned up before the data migration.

24. How are invoices generated?

Each employee manually does whatever they want for the invoices and itineraries.

25. What customer data is required for customer profiles? What data is optional?

    Each employee gathered their own data so everything needs to be made consistent.

26. Is the range of product category adequate or does it need an expansion?

    What is currently available is good.

27. How do you keep track of data currently?

    Sticky notes, excel, access.

Industry codes are not really necessary for them.

Day 2 Questions and Answers

1. Walk us through the process that each customer goes through.
   There's two types of customers. Business and Leisure. While businesses usually know what they want, leisure customers sometimes know, and other times don't know at all. Their job is to go to suppliers and get the right product and plan the trip. From the suppliers they get the confirmation number and commission payment. They put the package together, sometimes have a down payment/deposit. Final payment must be done before the trip, though sometimes the final payment is done on arrival. Prices that they receive/find are based off of experience of the agents.

2. Walk us through what each supplier goes through.
   The suppliers contact them through phone/email/website. They gather information – tax, commission available – for each product.
   The supplier detail is given every 6 months
   Though sometimes they find other suppliers that are not on the list that is given.

3. Walk us through what the commission specialist does.

   The commission specialist goes through all the agents. They figure out which commissions should be paid and compare with commissions that have been received.

   They then have to contact the supplier if the commission has not been paid or if the customer has not used the product.

   There's 5 types of commissions:

   > Booking commission = owing commissions
   >
   > Paying commission
   >
   > Outstanding/overdue = owing becomes outstanding the moment after 6months due date
   >
   > Invalid commission = where customers didn't go on the trip
   >
   > Canceled commission = cancelled product before the trip start

   The commission is at the discretion of the supplier for the rate and amount. It's individual for each product at the time of booking and not for each product.

   Price rates are per product and not per person.

   Taxes are based on the product.

   They must be able to handle multiple types of payment

Summary

Data has errors. Treat it as private, confidential data. Flag errors in data, some might be fixed before updating to the database. We want to be able to load the data and flag inconsistencies.

More information will be given later.

Try and keep data consistent.

Business and leisure travel clients.

Want to have a choice for if customer wants to give them credit card information. Therefore should be able to have the option to store this information.

Suppliers give paper or electronic version of their information. Its $75 per agent for paper, or $300 for electronic. They might want a way to just get one electronic version and be able to share it between all agents. Comes twice a year (every 6 months).

There are reward cards. Cards and names could be stored for the customers as well if they so desired.

Trips are made up of products.

Products must have a start and end date. These days could overlap between different products.

There's different classes of travel.

Prices are per product and not per person.

Supplier gives booking and confirmation number. This is only unique for the suppliers. But they also might get recycled within the suppliers products numbers.

Each invoice/itinerary is with a payment.

There's 3 types of payment. Full, deposit, and final. Could have more than one deposit.

Customers could make payment at the time of arrival/usage of product, such as car rentals.

If they charge agency fee ($25), it will always be on the first invoice, at the discretion of the agent.

Commissions are given by the supplier. Commission of the product is not stored in the database because it might change for the product each time it gets booked. Probably want to store it for the booking number.

Agents get a salary.

Commission type: owing, paid, overdue, invalid, and cancelled.

Insurance is also a product

Manager has full access. Commission specialist has full access. And hiring DBA.

They want to make sure they still have a copy of the data even if agents leave.

Hours: Mon-Sat = 9-18, Thurs = 9 -21, Sun = closed.

Unasked Questions

1. What would be a success for you?
2. Are there any obstacles that you know of currently that may prevent this project from being a success?
3. What is necessary?
4. What is nice to have?
5. What kind of data do you keep track of currently? What do you enter? What do you think should be automated?
6. What do you not keep track of?
7. What additional data would be helpful?
8. What do you think is a must have for this database? Website?
9. Is all the data supplied by each supplier consistent? Would we need to create a consistent form for all suppliers to fill out?
10. How is commission taxed?
11. On what basis is the commission % determined?
12. Who keeps track of commissions – how do they keep track? What do they need to keep track? What can they keep track of?
13. If the customer cancels the reservation, how would the refund be processed?

PROJECT DEFINITION

Project Start Date: October 28, 2019 Project End Date: May 1, 2020

Project Objectives:

Create a database to centralize data, maintain current data, and generate standardized forms with easy to use graphical user interface.

Approach:

PHASE	ACTIVITY	DELIVERABLES
Planning	Initiate Project	Interview Q&A Detailed Project Schedule
Analysis	Define Functional Requirements Define Data Requirements Define Technical Requirements Application Testing Strategy Data Conversion Strategy Training Strategy Go Live Strategy Transition Strategy	Business Requirements Document Preliminary Budget Document
Design	Design Application Design Database Design Technical Infrastructure Design Application Test Scenarios Design Data Conversion & Test Scenarios Design Training	ER Diagram Table Instance Charts (TICs) Updated Project tasks and Schedule Business System Design Document (ERD describing document)
Construction	Construct Application Modules Construct Database Construct Technical Infrastructure Develop Application Test Cases Develop Data Conversion Modules Develop Data Conversion Test Cases Develop Training Materials	Application Software Database Technical Infrastructure Test Cases & Schedule Data Conversion Software Data Conversion Test Cases Training Materials & Schedule

	Develop Transition Materials	Go Live Schedule
Testing	System Testing Team Testing – Application/Data Conv. User Acceptance Testing – Application/Data Conv. Final User Acceptance Testing	Accepted Application &b Database Accepted Data Conversion
Rollout	Go Live	Live Software & Database
Transition	Handover to Support Team	Operational Business System

SCOPE

In Scope:

Creating a database in Oracle.

Centralizing and standardizing data.

Standardizing itinerary and invoice forms.

Designing a network structure.

Setting up for future website integration with the database.

Optimizing database.

Create a training guide for future users for the system.

Design and set up a system to allow for easier tracking of employee commissions

Clean up data, and create a model to prevent future

Flag data after migrating if errors are present

Keep track of invoices that has been sent using the database

Keep track all the different products available using the database

Create a way for the agency be able to use one electronic copy of the supplier details for all their agents.

Out Scope:

Marketing to clients.

Creating a website.

Transferring data from a sticky note.

Provide the ISP services

BUSINESS REQUIREMENTS

Functional Requirements (End-users)

No.	Requirement	Mandatory / Would Like
1	Generate standardized forms for itinerary and invoices.	Mandatory
2	Insert data in a centralized location	Mandatory
3	A way to enable future website integration	Would like
4	A graphical interface for the database	Mandatory
5	A way to get commission system to work – store dates and specific commissions for each booking, and keep track of the status of the commissions.	Mandatory
6	Be able to have multiple users work using the same database at the same time.	Mandatory
7	Have backup data incase anything goes wrong	Mandatory
8	Keep track of customer credit card information in a safe, secure manner	Optional
9	Keep track of customer preferences, birthdays, anniversary days, food allergies, etc…	Optional
10	A way for all agents to share supplier information from single electronic copy	Optional
11	Allow agents to inquire about the different products from different suppliers.	Mandatory
12	Allow agents to look up customer histories for repeating trips	Mandatory
13	Standardize all data so it's easier for searching	Mandatory

Data Requirements

Information Type	Requirements
Access & storage needs	Require specific groups of people to have full access (DBA, commission manager, and manager)
Data growth expectation	Be able to handle at least double the size in the period of 2 years.
Data columns	Customer details, trip details, supplier details, product detail, commission due dates per product
Access & storage needs	Require centralized data
Access & storage needs	Store as much historical data as possible, for returning users and tax reasons.

Technical Infrastructure Requirements

No.	Requirement
1	20 desktops
2	Servers
3	Routers
4	Oracle
5	Backup/Recovery
6	Firewalls
7	Spam filter
8	Cables
9	File Storage Device (NAS)
10	Switches
11	Server Licenses

Preliminary Budget Breakdown

Hardware/Licensing Costs

Desktop: $20 000

Monitors: $3 000

Computer Accessories: $3 000

Routers: $660

Switches: $780

NAS: $3 600

Spam Filter: $3 000

Firewall: $960

Cables: $1 250

Servers: $14 400

Server Towers: $600

Linux: $2 600

Server Licensing: $15 300

Oracle Licensing: $190 000

Subtotal: $260 450

Human Resource

HR Costs Up to the First 2 Workshops: $3 000

Future HR Cost Estimate: $25 000

Subtotal: $28 000

Total Estimated Required Budget: $288 450

Left Over Budget: $211 910

Recommendations

For the operating system, we recommend the Red Hat OS. Linux base operating system is more secure and stable than Windows operating systems, and scripts are much easier to maintain and schedule using this system.

For the database, we suggest the Travel Experts Agency to obtain the licensing for Oracle. By utilizing the Oracle database, several additional features can be utilized, such as the Oracle Enterprise Manager (OEM) and Oracle Application Express (ApEx).

The first, OEM, will allow for easier analysis of the database performance for better performance and tuning adjustments. However, it does require extra licensing fee that currently is not included in the budget. In addition to the OEM, index creation on tables with constant searches, such as Product, TripProduct, Invoices are recommended as this will speed up the searches. As these tables do have a lot of changes, we also recommend daily rebuilding of the indexes using provided scripts to keep the running at optimal speeds.

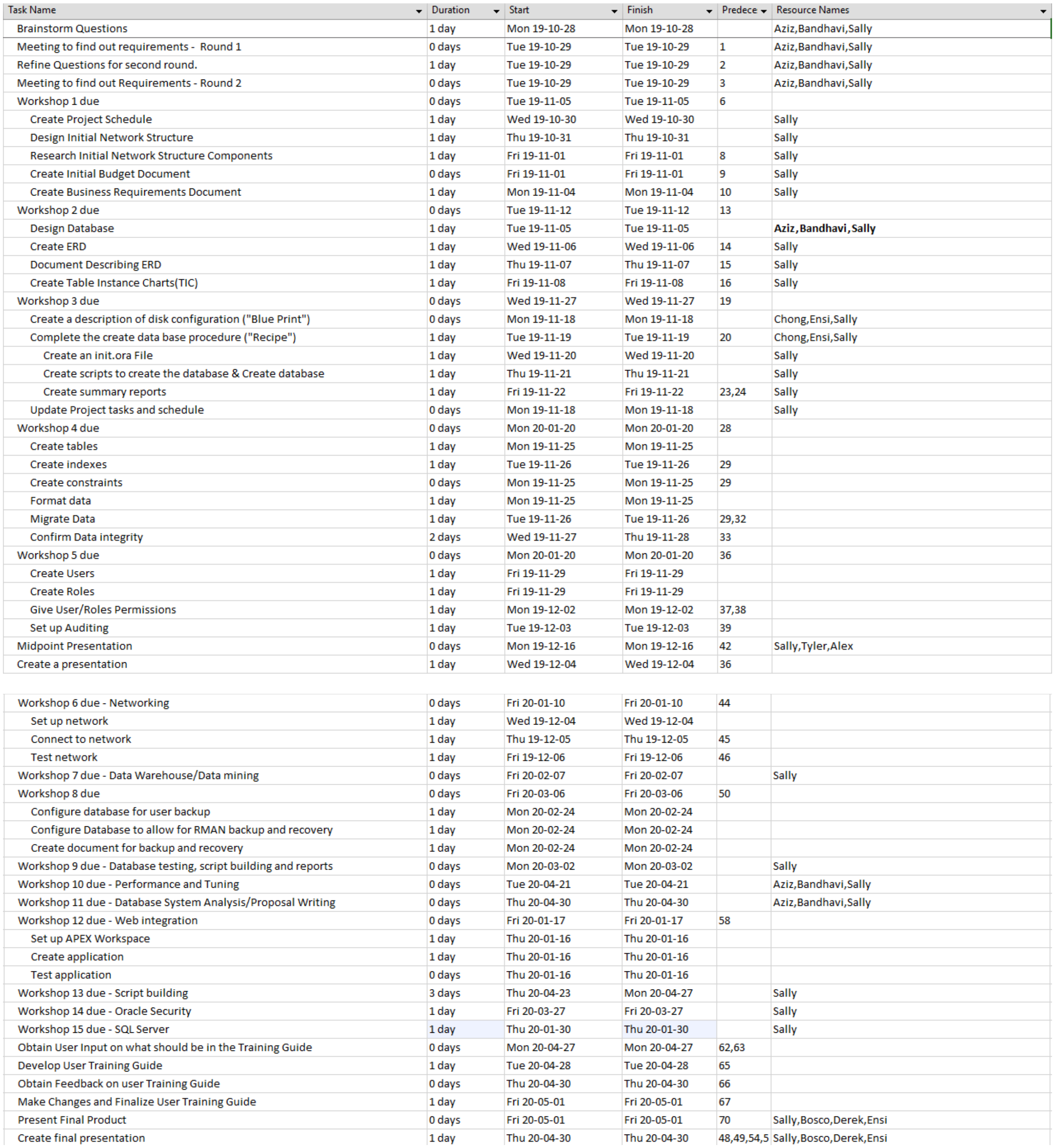
ApEx is a way for the database to be easily integrated into the web. It has a simple interface that is easily navigable and allow employees and customers alike to produce reports from anywhere with a network connection.

Performance and Tuning

While it is not possible at the current time to state what may need to be tuned to improve performance, certain things to watch for when the database is up and running are the redo log sizes and the execution times and %IO which will indicate badly formatted SQL.

While the current proposed database setup should not run into any hot blocks - congestion, if it is ever detected, increased buffer cache size is recommended.

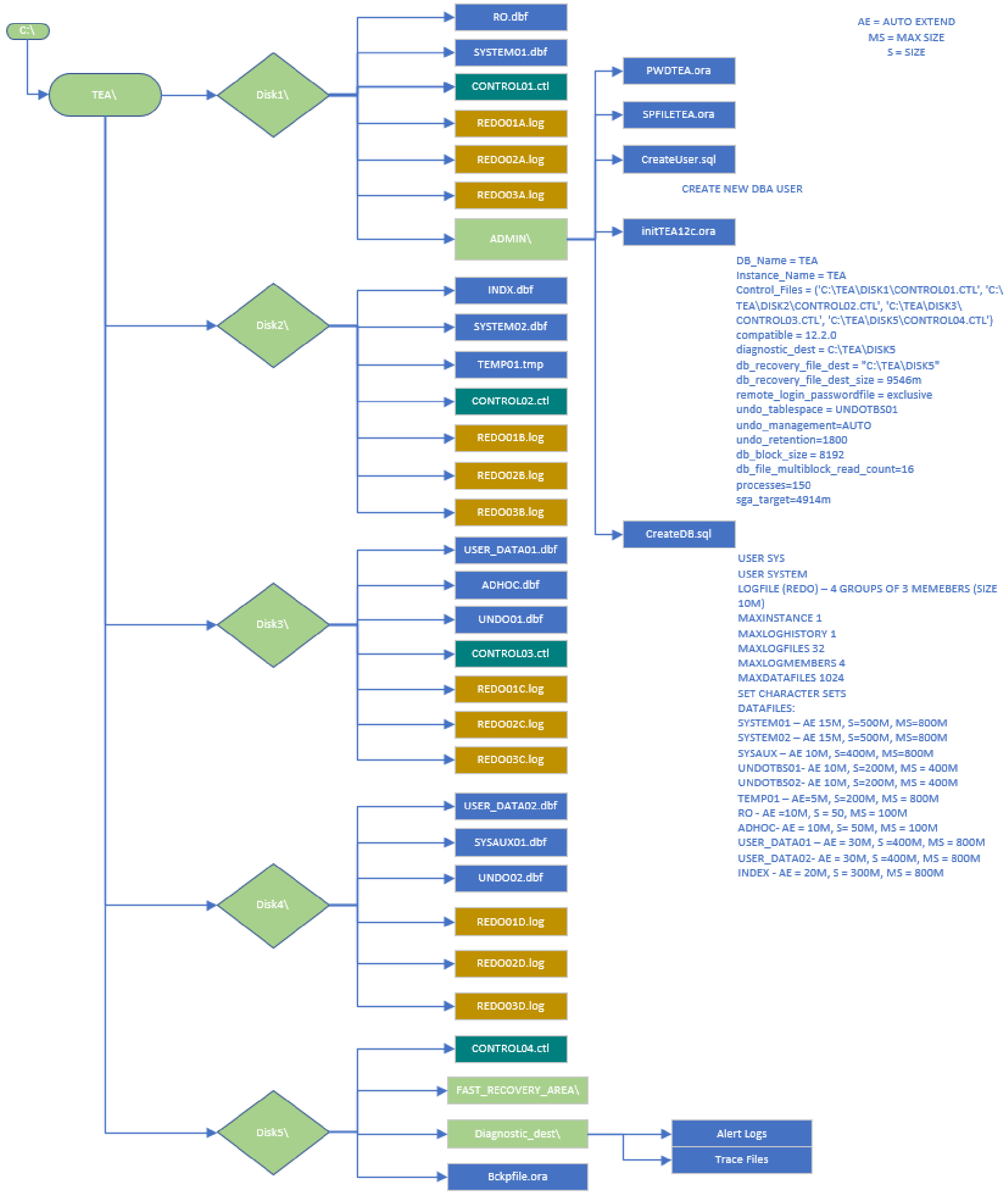
Schedule



Task Name	Duration	Start	Finish	Predece	Resource Names
Brainstorm Questions	1 day	Mon 19-10-28	Mon 19-10-28		Aziz,Bandhavi,Sally
Meeting to find out requirements - Round 1	0 days	Tue 19-10-29	Tue 19-10-29	1	Aziz,Bandhavi,Sally
Refine Questions for second round.	1 day	Tue 19-10-29	Tue 19-10-29	2	Aziz,Bandhavi,Sally
Meeting to find out Requirements - Round 2	0 days	Tue 19-10-29	Tue 19-10-29	3	Aziz,Bandhavi,Sally
Workshop 1 due	0 days	Tue 19-11-05	Tue 19-11-05	6	
Create Project Schedule	1 day	Wed 19-10-30	Wed 19-10-30		Sally
Design Initial Network Structure	1 day	Thu 19-10-31	Thu 19-10-31		Sally
Research Initial Network Structure Components	1 day	Fri 19-11-01	Fri 19-11-01	8	Sally
Create Initial Budget Document	0 days	Fri 19-11-01	Fri 19-11-01	9	Sally
Create Business Requirements Document	1 day	Mon 19-11-04	Mon 19-11-04	10	Sally
Workshop 2 due	0 days	Tue 19-11-12	Tue 19-11-12	13	
Design Database	1 day	Tue 19-11-05	Tue 19-11-05		**Aziz,Bandhavi,Sally**
Create ERD	1 day	Wed 19-11-06	Wed 19-11-06	14	Sally
Document Describing ERD	1 day	Thu 19-11-07	Thu 19-11-07	15	Sally
Create Table Instance Charts(TIC)	1 day	Fri 19-11-08	Fri 19-11-08	16	Sally
Workshop 3 due	0 days	Wed 19-11-27	Wed 19-11-27	19	
Create a description of disk configuration ("Blue Print")	0 days	Mon 19-11-18	Mon 19-11-18		Chong,Ensi,Sally
Complete the create data base procedure ("Recipe")	1 day	Tue 19-11-19	Tue 19-11-19	20	Chong,Ensi,Sally
Create an init.ora File	1 day	Wed 19-11-20	Wed 19-11-20		Sally
Create scripts to create the database & Create database	1 day	Thu 19-11-21	Thu 19-11-21		Sally
Create summary reports	1 day	Fri 19-11-22	Fri 19-11-22	23,24	Sally
Update Project tasks and schedule	0 days	Mon 19-11-18	Mon 19-11-18		Sally
Workshop 4 due	0 days	Mon 20-01-20	Mon 20-01-20	28	
Create tables	1 day	Mon 19-11-25	Mon 19-11-25		
Create indexes	1 day	Tue 19-11-26	Tue 19-11-26	29	
Create constraints	0 days	Mon 19-11-25	Mon 19-11-25	29	
Format data	1 day	Mon 19-11-25	Mon 19-11-25		
Migrate Data	1 day	Tue 19-11-26	Tue 19-11-26	29,32	
Confirm Data integrity	2 days	Wed 19-11-27	Thu 19-11-28	33	
Workshop 5 due	0 days	Mon 20-01-20	Mon 20-01-20	36	
Create Users	1 day	Fri 19-11-29	Fri 19-11-29		
Create Roles	1 day	Fri 19-11-29	Fri 19-11-29		
Give User/Roles Permissions	1 day	Mon 19-12-02	Mon 19-12-02	37,38	
Set up Auditing	1 day	Tue 19-12-03	Tue 19-12-03	39	
Midpoint Presentation	0 days	Mon 19-12-16	Mon 19-12-16	42	Sally,Tyler,Alex
Create a presentation	1 day	Wed 19-12-04	Wed 19-12-04	36	
Workshop 6 due - Networking	0 days	Fri 20-01-10	Fri 20-01-10	44	
Set up network	1 day	Wed 19-12-04	Wed 19-12-04		
Connect to network	1 day	Thu 19-12-05	Thu 19-12-05	45	
Test network	1 day	Fri 19-12-06	Fri 19-12-06	46	
Workshop 7 due - Data Warehouse/Data mining	0 days	Fri 20-02-07	Fri 20-02-07		Sally
Workshop 8 due	0 days	Fri 20-03-06	Fri 20-03-06	50	
Configure database for user backup	1 day	Mon 20-02-24	Mon 20-02-24		
Configure Database to allow for RMAN backup and recovery	1 day	Mon 20-02-24	Mon 20-02-24		
Create document for backup and recovery	1 day	Mon 20-02-24	Mon 20-02-24		
Workshop 9 due - Database testing, script building and reports	0 days	Mon 20-03-02	Mon 20-03-02		Sally
Workshop 10 due - Performance and Tuning	0 days	Tue 20-04-21	Tue 20-04-21		Aziz,Bandhavi,Sally
Workshop 11 due - Database System Analysis/Proposal Writing	0 days	Thu 20-04-30	Thu 20-04-30		Aziz,Bandhavi,Sally
Workshop 12 due - Web integration	0 days	Fri 20-01-17	Fri 20-01-17	58	
Set up APEX Workspace	1 day	Thu 20-01-16	Thu 20-01-16		
Create application	1 day	Thu 20-01-16	Thu 20-01-16		
Test application	0 days	Thu 20-01-16	Thu 20-01-16		
Workshop 13 due - Script building	3 days	Thu 20-04-23	Mon 20-04-27		Sally
Workshop 14 due - Oracle Security	1 day	Fri 20-03-27	Fri 20-03-27		Sally
Workshop 15 due - SQL Server	1 day	Thu 20-01-30	Thu 20-01-30		Sally
Obtain User Input on what should be in the Training Guide	0 days	Mon 20-04-27	Mon 20-04-27	62,63	
Develop User Training Guide	1 day	Tue 20-04-28	Tue 20-04-28	65	
Obtain Feedback on user Training Guide	0 days	Thu 20-04-30	Thu 20-04-30	66	
Make Changes and Finalize User Training Guide	1 day	Fri 20-05-01	Fri 20-05-01	67	
Present Final Product	0 days	Fri 20-05-01	Fri 20-05-01	70	Sally,Bosco,Derek,Ensi
Create final presentation	1 day	Thu 20-04-30	Thu 20-04-30	48,49,54,5	Sally,Bosco,Derek,Ensi

Database Specifics

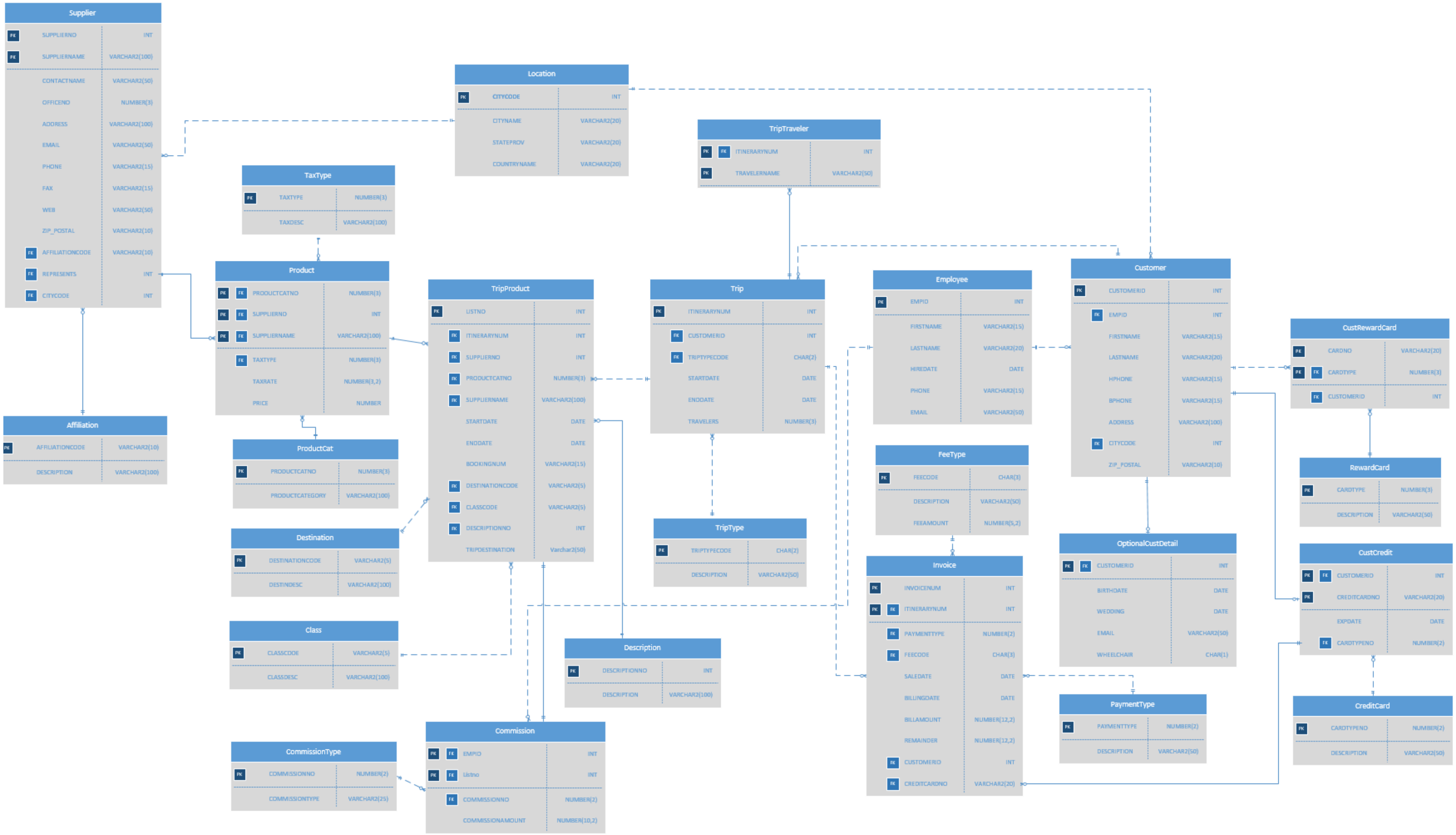
Database Blueprint



```
DB_Name = TEA
Instance_Name = TEA
Control_Files = ('C:\TEA\DISK1\CONTROL01.CTL', 'C:\TEA\DISK2\CONTROL02.CTL', 'C:\TEA\DISK3\CONTROL03.CTL', 'C:\TEA\DISK5\CONTROL04.CTL')
compatible = 12.2.0
diagnostic_dest = C:\TEA\DISK5
db_recovery_file_dest = "C:\TEA\DISK5"
db_recovery_file_dest_size = 9546m
remote_login_passwordfile = exclusive
undo_tablespace = UNDOTBS01
undo_management=AUTO
undo_retention=1800
db_block_size = 8192
db_file_multiblock_read_count=16
processes=150
sga_target=4914m
```

USER SYS
USER SYSTEM
LOGFILE (REDO) – 4 GROUPS OF 3 MEMEBERS (SIZE 10M)
MAXINSTANCE 1
MAXLOGHISTORY 1
MAXLOGFILES 32
MAXLOGMEMBERS 4
MAXDATAFILES 1024
SET CHARACTER SETS
DATAFILES:
SYSTEM01 – AE 15M, S=500M, MS=800M
SYSTEM02 – AE 15M, S=500M, MS=800M
SYSAUX – AE 10M, S=400M, MS=800M
UNDOTBS01- AE 10M, S=200M, MS = 400M
UNDOTBS02- AE 10M, S=200M, MS = 400M
TEMP01 – AE=5M, S=200M, MS = 800M
RO - AE =10M, S = 50, MS = 100M
ADHOC- AE = 10M, S= 50M, MS = 100M
USER_DATA01 – AE = 30M, S =400M, MS = 800M
USER_DATA02- AE = 30M, S =400M, MS = 800M
INDEX - AE = 20M, S = 300M, MS = 800M

ERD


Supplier
PK SUPPLIERNO INT
PK SUPPLIERNAME VARCHAR2(100)
CONTACTNAME VARCHAR2(50)
OFFICENO NUMBER(3)
ADDRESS VARCHAR2(100)
EMAIL VARCHAR2(50)
PHONE VARCHAR2(15)
FAX VARCHAR2(15)
WEB VARCHAR2(50)
ZIP_POSTAL VARCHAR2(10)
FK AFFILIATIONCODE VARCHAR2(10)
FK REPRESENTS INT
FK CITYCODE INT
Affiliation
PK AFFILIATIONCODE VARCHAR2(10)
DESCRIPTION VARCHAR2(100)
TaxType
PK TAXTYPE NUMBER(3)
TAXDESC VARCHAR2(100)
Product
PK FK PRODUCTCATNO NUMBER(3)
PK FK SUPPLIERNO INT
PK FK SUPPLIERNAME VARCHAR2(100)
FK TAXTYPE NUMBER(3)
TAXRATE NUMBER(3,2)
PRICE NUMBER
ProductCat
PK PRODUCTCATNO NUMBER(3)
PRODUCTCATEGORY VARCHAR2(100)
Destination
PK DESTINATIONCODE VARCHAR2(5)
DESTINDESC VARCHAR2(100)
Class
PK CLASSCODE VARCHAR2(5)
CLASSDESC VARCHAR2(100)
CommissionType
PK COMMISSIONNO NUMBER(2)
COMMISSIONTYPE VARCHAR2(25)
Location
PK CITYCODE INT
CITYNAME VARCHAR2(20)
STATEPROV VARCHAR2(20)
COUNTRYNAME VARCHAR2(20)
TripProduct
PK LISTNO INT
FK ITINERARYNUM INT
FK SUPPLIERNO INT
FK PRODUCTCATNO NUMBER(3)
FK SUPPLIERNAME VARCHAR2(100)
STARTDATE DATE
ENDDATE DATE
BOOKINGNUM VARCHAR2(15)
FK DESTINATIONCODE VARCHAR2(5)
FK CLASSCODE VARCHAR2(5)
FK DESCRIPTIONNO INT
TRIPDESTINATION Varchar2(50)
Commission
PK FK EMPID INT
PK FK Listno INT
FK COMMISSIONNO NUMBER(2)
COMMISSIONAMOUNT NUMBER(10,2)
Description
PK DESCRIPTIONNO INT
DESCRIPTION VARCHAR2(100)
TripTraveler
PK FK ITINERARYNUM INT
PK TRAVELERNAME VARCHAR2(50)
Trip
PK ITINERARYNUM INT
FK CUSTOMERID INT
FK TRIPTYPECODE CHAR(2)
STARTDATE DATE
ENDDATE DATE
TRAVELERS NUMBER(3)
TripType
PK TRIPTYPECODE CHAR(2)
DESCRIPTION VARCHAR2(50)
Employee
PK EMPID INT
FIRSTNAME VARCHAR2(15)
LASTNAME VARCHAR2(20)
HIREDATE DATE
PHONE VARCHAR2(15)
EMAIL VARCHAR2(50)
FeeType
PK FEECODE CHAR(3)
DESCRIPTION VARCHAR2(50)
FEEAMOUNT NUMBER(5,2)
Invoice
PK INVOICENUM INT
PK FK ITINERARYNUM INT
FK PAYMENTTYPE NUMBER(2)
FK FEECODE CHAR(3)
SALEDATE DATE
BILLINGDATE DATE
BILLAMOUNT NUMBER(12,2)
REMAINDER NUMBER(12,2)
FK CUSTOMERID INT
FK CREDITCARDNO VARCHAR2(20)
Customer
PK CUSTOMERID INT
FK EMPID INT
FIRSTNAME VARCHAR2(15)
LASTNAME VARCHAR2(20)
HPHONE VARCHAR2(15)
BPHONE VARCHAR2(15)
ADDRESS VARCHAR2(100)
FK CITYCODE INT
ZIP_POSTAL VARCHAR2(10)
OptionalCustDetail
PK FK CUSTOMERID INT
BIRTHDATE DATE
WEDDING DATE
EMAIL VARCHAR2(50)
WHEELCHAIR CHAR(1)
PaymentType
PK PAYMENTTYPE NUMBER(2)
DESCRIPTION VARCHAR2(50)
CustRewardCard
PK CARDNO VARCHAR2(20)
PK FK CARDTYPE NUMBER(3)
FK CUSTOMERID INT
RewardCard
PK CARDTYPE NUMBER(3)
DESCRIPTION VARCHAR2(50)
CustCredit
PK FK CUSTOMERID INT
PK CREDITCARDNO VARCHAR2(20)
EXPDATE DATE
FK CARDTYPENO NUMBER(2)
CreditCard
PK CARDTYPENO NUMBER(2)
DESCRIPTION VARCHAR2(50)

Parameter File

```
tea.__data_transfer_cache_size=0
tea.__db_cache_size=3825205248
tea.__java_pool_size=50331648
tea.__large_pool_size=117440512
tea.__oracle_base='D:\app\oracle'#ORACLE_BASE set from environment
tea.__pga_aggregate_target=1040187392
tea.__sga_target=5167382528
tea.__shared_io_pool_size=268435456
tea.__shared_pool_size=889192448
tea.__streams_pool_size=0
*.AUDIT_FILE_DEST='c:\tea\disk5'
*.audit_trail='db','extended'
*.compatible='12.0.0'
*.Control_Files='C:\TEA\DISK1\CONTROL01.CTL','C:\TEA\DISK2\CONTROL02.CTL','C
:\TEA\DISK3\CONTROL03.CTL','C:\TEA\DISK5\CONTROL04.CTL'
*.db_block_size=8192
*.db_file_multiblock_read_count=16
*.DB_Name='TEA'
*.db_recovery_file_dest='C:\TEA\DISK5'
*.db_recovery_file_dest_size=9546m
*.diagnostic_dest='C:\TEA\DISK5'
*.Instance_Name='TEA'
*.processes=150
*.remote_login_passwordfile='exclusive'
*.sga_target=4914m
*.undo_management='AUTO'
*.undo_retention=1800
*.undo_tablespace='UNDOTBS01'
```

Network Files

TNSNAMES

```
# tnsnames.ora Network Configuration File:
C:\app\Administrator\product\12.2.0\dbhome_1\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

LISTENER_ORANT12C =
  (ADDRESS = (PROTOCOL = TCP)(HOST = oralocal)(PORT = 1521))

ORANT12C =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = oralocal)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orant12c)
    )
  )
ORACLR_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
    (CONNECT_DATA =
      (SID = CLRExtProc)
      (PRESENTATION = RO)
    )
  )
ORCLPDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = oralocal)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orclpdb)
    )
  )
NETCLASS =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = oralocal)(PORT = 1423))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = NETCLASS)
    )
  )
TEA =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ICTVM-OBOM100Q3.ACDM.DS.SAIT.CA)(PORT = 
1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = tea)
    )
  )
```

LISTENER

```
C:\app\Administrator\product\12.2.0\dbhome_1\network\admin\listener.ora
# Generated by Oracle configuration tools.

LISTENER2 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ICTVM-O09KF103D)(PORT = 1522))
  )
ADR_BASE_NETLISTENER = C:\app\Administrator\product\12.2.0\dbhome_1\log

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = C:\app\Administrator\product\12.2.0\dbhome_1)
      (PROGRAM = extproc)
      (ENVS = 
"EXTPROC_DLLS=ONLY:C:\app\Administrator\product\12.2.0\dbhome_1\bin\oraclr12.dl
l")
    )
  )
```

```
SID_LIST_LISTENER2 =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = TEA)
      (SID_NAME = TEA)
    )
  )
ADR_BASE_LISTENER2 = C:\app\Administrator\product\12.2.0\dbhome_1\log

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = oralocal)(PORT = 1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )
ADR_BASE_LISTENER = C:\app\Administrator\product\12.2.0\dbhome_1\log

NETLISTENER =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ICTVM-O09KF103D)(PORT = 1423))
  )
SID_LIST_NETLISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = NETCLASS)
      (SID_NAME = NETCLASS)
    )
  )
```

SQLNET

```
C:\app\Administrator\product\12.2.0\dbhome_1\network\admin\sqlnet.ora
# Generated by Oracle configuration tools.

# This file is actually generated by netca. But if customers choose to
# install "Software Only", this file wont exist and without the native
# authentication, they will not be able to connect to the database on NT.

SQLNET.AUTHENTICATION_SERVICES= (NTS)

NAMES.DIRECTORY_PATH= (TNSNAMES, HOSTNAME, EZCONNECT)

ADR_BASE = C:\app\Administrator\product\12.2.0\dbhome_1\log
```

Security

Database Locking

- Need to keep track of:
  - Who is adding data
  - Who is changing data
  - What is the security level of the record (0 = everyone can see. 99 = top?)
- Never add columns to existing tables, it will break many different programs/scripts
  - So what you do is beside every parent table in the database, you create a 1 to 1 child table

1) Create a 1 to 1 table for each parent table, however do not make it a foreign key, so that any deletion of the table doesn't delete this security table
2) What are the security levels of the data
3) Create user security
4) Start at the bottom level and add more levels
5) Always authorize to views and not the tables directly.
6) Also need procedures on each table, ex select 1 row. Select all rows. Insert, update delete.
7) Security log table

Steps needed to secure each table:

1) Make user security table of users and security level
   - Only need one for database
2) Security log giving security level for each primary key
   - Normally for parent tables only
3) Security log details for each table to track every select, insert, update and delete
4) View linking parent table to security log and to user security
5) Triggers to update security logs for insert, update, and delete
6) Procedures to replace select, insert, update, delete
   - Procedure for insert, update, delete, select all rows, select one row
7) Revoke privilege to all tables, and only allow views and procedures.

Backup

Goal

The purpose of the backup is to create a copy of data that can be recovered in the event of a data failure.
Data failure could be a software or a hardware failure or user errors.
Having multiple consistent backup copies minimize data loss and protect the agency's primary data from corruption, users-errors or any malicious attacks.

Our hope with this document is to create a plan and lay down strategies that the Travel Expert Agency can follow to have a database that has the highest uptime and little downtime and with little to no data loss if the database services are ever interrupted due to errors.

Service Level Agreement (SLA)

The Travel Expert Agency database will be backed up with a mixture of user managed, RMAN and logical backup methods that are further explained in this document. The database is expected to be online for all but 2 hour per week. Further details on how the backups will be scheduled is stated later on in this document. The specification of the database is stated in the user-managed configurations.

To increase the mean time between failures, multiplexing of the files will be implemented.

Encryption is currently not part of the database backups.

Will require agreement from all management levels about what this backup and recovery plan will guarantee.

The read-only data will only be backed up when they are updated. This is to reduce the performance hit, but skipping the redundant backups.

UPS

Uninterruptible Power supply (UPS) – The database is initially set to not have this feature, however it can be implemented later on when the company grows further.

Contacts

In order to maintain a good backup practice, must keep track of the contact information of all personal who maintains the database and deals with the long term tape storage.

User-Managed

Configurations

Physical storages

5 different disks:

DISK1
DISK2
DISK3
DISK4
DISK5

Parameters

Parameter	**Value**
DB_CREATE_FILE_DEST	C:\TEA\DISK5
DB_FLASHBACK_RETENTION_TARGET	1440
UNDO_TABLESPACE	UNDOTBS01
UNDO_MANAGEMENT	AUTO
UNDO_RETENTION	1800
DB_RECOVERY_FILE_DEST	“C:\TEA\DISK5”
DB_RECOVERY_FILE_DEST_SIZE	9546m
DIAGNOSTIC_DEST	C:\TEA\DISK5
LOG_ARCHIVE_DEST_1	"LOCATION=c:\tea\disk2\archive"
LOG_ARCHIVE_DEST_2	"LOCATION=c:\tea\disk3\archive"
LOG_ARCHIVE_DEST_3	"LOCATION=c:\tea\disk4\archive"
COMPATIBLE	12.0.0
SGA_TARGET	4914m
LOG_ARCHIVE_FORMAT	'tea_log_%s_%t_%r.arc'
CONTROLFILE	('C:\TEA\DISK1\CONTROL01.CTL', 'C:\TEA\DISK2\CONTROL02.CTL', 'C:\TEA\DISK3\CONTROL03.CTL', 'C:\TEA\DISK5\CONTROL04.CTL')
CONTROL_FILE_RECORD_KEEP_TIME	30
FAST_START_MTTR_TARGET	180

File Configurations

Control Files

Multiplexing control files, 4 on different disks.

C:\TEA\DISK1\CONTROL01.CTL
C:\TEA\DISK2\CONTROL02.CTL
C:\TEA\DISK3\CONTROL03.CTL
C:\TEA\DISK5\CONTROL04.CTL

Redo Logs

8 groups, each group has 3 members

First member in each group:

C:\TEA\DISK1\redona.log

Second member in each group:

C:\TEA\DISK2\redonb.log

Third member in each group:

C:\TEA\DISK3\redonc.log

Where n = group number (format = 01)

Data Files

Adhoc: C:\TEA\DISK3\adhoc.dbf
System: C:\TEA\DISK1\system01.dbfTemp:
Indx: C:\TEA\DISK2\indx.dbf
SysAux: C:\TEA\Disk4\sysaux01.dbf
UserData: C:\TEA\DISK3\user_data01.dbf, C:\TEA\DISK4\user_data02.dbf
ReadOnly: C:\TEA\DISK1\ro.dbf
Undo: C:\TEA\DISK3\undo01.dbf
Temp: C:\TEA\Disk2\temp01.dbf

Tablespaces

To make updating easier, the data should be separated into different tablespaces based on their volatility. Those with little to no change, such as the read only tablespaces, do not require multiple backups weekly; they would only need to be backed up when they are changed, while the tablespaces such as invoices that are changed throughout the day would require more backups so less data is lost and reduce the mean time to recover.

Long Term Tape Storage

Have a long term back up multiple copies on tapes. Keep multiple (at least 2 copies) copies of your backup in different and safe locations, easy place to reach.

7 tapes for incremental daily backup. Reuse them every week

4 tapes for weekly backup. Reuse them every 4 weeks

12 tapes for monthly backup. Reuse them every year

7 tapes for 7 years backup (according to the law). Reuse them every 7 years

Backup Locations:

Travel agency office

SAIT

Iron Mountain

Archive log mode

To make physical copies of online filled redo log files to one or multiple locations before they get overwritten.

1. To check the log archive destinations
2. For backing up the last changes
3. Shut down properly
4. Startup in mount stage
5. Enable archive log mode
6. set in the Pfile the needed destination in different disks, and set the format of the pfile
7. start up the system
8. verify the destinations
9. to check the status of the back up
10. open the database

```
archive log list;
Alter system archive log current;
shutdown normal
startup mount pfile=C:\tea\disk1\admin\initTEA12C.ora;
alter database archivelog;
--an archival destination can be disable by using
pfile: log_archive_dest_state_1=defer
or
alter system set log_archive_dest_state_1=defer;
--enable archive to a destination again:
pfile: log_archive_dest_state_1=enable
or
alter system set log_archive_dest_state_1=enable;

startup pfile=C:\tea\disk1\admin\initTEA12c.ora
Show parameter log_archive;
select * from v$backup;
```

```
alter database open;
```

Hot Backup

Hot backups are performed while database is opened and available for users. It requires the database to be in archive log

Advantages:

Maintains high db availability
Can be done at a tablespace or datafile level
Supports nonstop business operations

1. relocate a redo log file
2. Get the location of the required tablespace
3. Mark the header to not update and change anything in the log by run:
4. Copy the datafiles and Specify the target and the needed destination:
5. Verify backup status information (which files are in backup mode)
6. End backup

```
Alter system archive log current;
Or
Alter system switch logfile;
Select tablespace_name, file_name from dba_data_files;
Alter tablespace tbsname begin backup;
copy C:\tea\...\.rdo       C:\tea\...\.rdo
select * from v$backup;          --status will be "active"
Alter tablespace tbsname end backup;
select * from v$backup;          --status will be "not active"
     alter system archive log current;
```

Make sure that the files were put in backup mode, not restored from a backup

```
desc v$datafile_header
select file#, status, chec kpoint_time from v$datafile_header;
```

Logical Backup

Logical backups are done via export and import using the datapump utility. It used to backup precise object such Specific table from a specific schema.

To reduce the amount of possible data loss, tablespaces, indirectly the data files, will be categorized by the data volatility and exported using Oracle datapump utility according to the schedule.

Control Files

Manual

Multiplexing, create copies of the control file in different disks – Cold Backup.

Binary File

Create a duplicate binary file

SQL> ALTER DATABASE BACKUP CONTROLFILE TO ‘C:\tea\control.bkp’;

Trace

1. Backup control file using Trace
2. Look through the alert log to figure out the location

SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE;

Recovery Manager (RMAN)

A backup and recovery manager that will be used to back up the Travel Experts Agency database.

Configurations

Parameters

Parameter	Value
RETENTION POLICY TO REDUNDANCY	2
BACKUP OPTIMIZATION	ON
DEFAULT DEVICE TYPE	DISK
CONTROLFILE AUTOBACKUP	ON
CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO	'%F'
DEVICE TYPE DISK PARALLELISM	3
RMAN OUTPUT TO KEEP	7

Recovery Catalog

Creating Catalog

1. Set the DB_RECOVERY_FILE_DEST, DB_RECOVERY_FILE_DEST_SIZE, DB_RECOVERY_FILE_DEST_N in the pfile
2. Set the instance for the required database to create a catalog
   - a different database will contain the catalog and not the TEA database
3. Verify the instance
4. In sqlplus connect to sys to create a recovery tablespace
5. Create the recovery catalog owner
6. Grant the required privileges to the catalog owner
7. Log in to rman connecting to the rcat_owner
8. Create the catalog

```
DB_RECOVERY_FILE_DEST= C:\BUR\F\recovery_bur"
DB_RECOVERY_FILE_DEST_SIZE = 9500M
C:\> Set oracle_sid=rcat
C:\> Set oracle_sid
SQL> conn sys as sysdba
SQL> CREATE TABLESPACE rcat_tbs DATAFILE…
SQL> CRAETE USER rcat_owner IDENTIFIED BY catowner
      DEFAULT TABLESPACE rcat_tbs
      QUOTA unlimited on rcat_tbs;
SQL> GRANT recovery_catalog_owner to rcat_owner;
SQL> GRANT CONNECT, RESOURCE TO rcat_owner;
C:\> rman catalog rcat_owner@rcat
RMAN> CREATE CATALOG;
```

Database Registering

1. Open a different command prompt window to set up the instance for the travel express database
2. Verify the instance
3. Connect to the target database (travel express) to register
4. Register the database in the catalog

```
C:\> Set oracle_sid=bur
C:\> Set oracle_sid
C:\> rman target sys@bur catalog rcat_owner@rcat
```

```
          RMAN> register database;
```

Incremental backup for the catalog database

For an incremental, backup the catalog database as an image copy

1. Log in RMAN using the catalog database as a target with no catalog
2. Backup the catalog database incrementally

```
C:\> RMAN target sys@rcat
RMAN> BACKUP AS COPY INCREMENTAL LEVEL 0 DATABASE;
```

Resync the recovery catalog

To manually resync the recovery catalog, connect to the target database and the catalog database, then resync

```
RMAN> connect target sys/Elcaro1!
RMAN>connect catalog rcatowner/rcat
RMAN>resync catalog;
```

RMAN Backup

Setting up the database

1. Mount the database
2. Set the database to archive log mode
3. Open the database
4. Verify that the database in the archive log mode

```
          SQL> SHUTDOWN IMMEDIATE
          SQL> STARTUP MOUNT pfile= C:\tea\ initTEA.ora
          SQL> ALTER DATABASE ARCHIVELOG;
          SQL> ALTER DATABASE OPEN;
          SQL> ARCHIVE LOG LIST
```

To specify the default location for the FRA set the DB_RCOVERY_FILE_DEST parameter in the Pfile

Multi-section Backups

For backups of large files; all sections except for the last are guaranteed to be the same size, maximum of 256 sections per file

```
backup <options> section size <integer> [K/M/G]
validate datafile <options> section size <integer> [K/M/G]
```

The multi_section column indicates whether this is a multi-section backup or not

v$backup_set
rc_backup_set --'rc_' -- when u create the catalog

The section_size column specifies the number of blocks in each section of a multi-section backup, "0" means a whole-file backup

v$backup_datafile
rc_backup_datafile

RMAN Backup Execution

For current control file backup:

```
          RMAN> backup current controlfile;
```

1. Log into RMAN to the target database
2. Backup the database

```
          C:\> rman target=sys@bur
          RMAN> BACKUP DATABASE;
```

Control and SP file will automatically be backed up if we set the autobackup on

1. Verify that the controlfile autobackup is on or off
2. Set the controlfile autobackup on if it is off

```
RMAN> Show controlfile autobackup;
RMAN> Configure controlfile autobackup on;
```

Incremental Backup

1. Backup the database to the required (non-default) directory
   - Allocate a channel to use the needed directory for the data files copies destination
2. Perform incremental level 1 for backup the new records and changes
   - We can also specify a channel and specify a name for this backup

```
RMAN> run {
          ALLOCATE CHANNEL "ch2" DEVICE TYPE DISK;
     BACKUP AS COPY TAG 'teabu01' INCREMENTAL LEVEL 0 DATABASE;}
RMAN> run {
          ALLOCATE CHANNEL "ch1" DEVICE TYPE DISK;
     BACKUP AS COPY TAG 'tea_update1' INCREMENTAL LEVEL 1 DATABASE;}
```

To recover the database using an incremental backup

```
RMAN> run {
     ALLOCATE CHANNEL "ch2" DEVICE TYPE DISK FORMAT
     "c:bur…";
     RECOVER COPY OF DATABASE WITH TAG 'teabu01';
```

Proxy Copies

```
     backup [as backupset] ... proxy [only] database/tablespace...
     desc v$proxy_datafile
-create duplexed backup sets by using "backup copies"
     RMAN>backup as backupset device type <sbt>
          copies 2
          incremental level 0
          database;
          list backup; --verify your backup
--create backups of backup sets
     RMAN> backup device type disk as backupset
          database plus archivelog;
          backup device type <sbt> backupset all;
```

1. Create restore point
2. Log in to RMAN using the catalog
3. Mount the database
4. Create an archival backup using a FORMAT clause that creates in the required and non-default destination and format because the archival backup has a big size to be stored in the FRA
5. Open the database

```
C:\> rman target sys catalog rcat_owner@rcat
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP MOUNT Pfile= C:\rcat\ rcatpfile.ora
```

```
RMAN> BACKUP DATABASE FORMAT ‘c:/rcat/F/BU/arcbu
        TAG arch_bu1
        KEEP forever RESTORE POINT tea_arch1;
SQL> ALTER DATABASE OPEN;
```

Verify Backup

Backups should be always verified right after backup session.
There are several ways to achieve this.

DBVERIFY

Checks if the backup is valid and may use in offline or online databases. (a page = a block)
In host, run the DBVerify utility to verify the files.

```
dbv help=y
dbv file= C:\TEA\DISK1\system01.dbf
dbv file= C:\TEA\DISK1\CONTROL01.CTL blocksize=16384
```

Manual

For Files such as the network files and pfile, which are non-binary files, you can manually open the backup files in a notepad or text editor to confirm that the correct values are present.

RMAN

In RMAN, verify the backup files.

```
RMAN> BACKUP VALIDATE CHECK LOGICAL DATABASE ARCHIVELOG ALL;
```

Backup Checklist

Things to keep track of for the backup:

1. Contact List
2. When backups were completed
3. Last Update to this backup plan
4. Last Full cold backup
5. List of all the data files and when and where they were backed up
6. Last time the catalog was resynced
7. Last time the backup was practiced
8. Last time updates to scripts were done
9. Last time someone updated the long term tapes for storage

Checklist Template

Example checklist template that can be filled up based on the schedule.

Who	What was achieved	When completed	Where is the backup saved?	Validity of backup
Sally	Backed up Files: Pfile SPFile Datafiles Control File	Mar. 6, 2020 at 12pm	Local backup	File validity confirmed
Chong	Files: Network Files (TNSname, Listener, SQLNet): Password File	Mar. 6, 2020 at 2pm	Tape backups	Pending…
Rami	Backed up the log files (Archive, Redo, alert) and cleared out old	Mar. 6, 2020 at 4pm	Local backups	File validity confirmed

Example Backup Schedule

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
	1 *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indexes, make sure the index creation script is updated* *13-15: Update documentations needed for backup* *12-13, 12-13, 23-24: Export Volatile data*	**2** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 12-13, 23-24: Export Volatile data*	**3** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 12-13, 23-24: Export Volatile data*	**4** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indexes, make sure the index creation script is updated* *12-13, 12-13, 23-24: Export Volatile data*	**5** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* ***14-15: Practice backup procedures in testing environment*** *12-13, 12-13, 23-24: Export Volatile data*	**6** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 12-13, 23-24: Export Volatile data*
7 *1-3: Full Cold Backup* *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 12-13, 23-24: Export Volatile data*	**8** 6-7: Export low volatile data *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *12-13, 12-13, 23-24: Export Volatile data*	**9** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *12-13, 12-13, 23-24: Export Volatile data*	**10** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *12-13, 12-13, 23-24: Export Volatile data*	**11** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *12-13, 12-13, 23-24: Export Volatile data*	**12** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *13-14: Review the backup plan for any updates needed* ***14-15: Practice backup procedures in testing environment*** *12-13, 12-13, 23-24: Export Volatile data*	**13** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *12-13, 12-13, 23-24: Export Volatile data*
14 *1-3: Full Cold Backup* *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the*	**15** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated*	**16** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated*	**17** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated*	**18** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated*	**19** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated*	**20** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated*

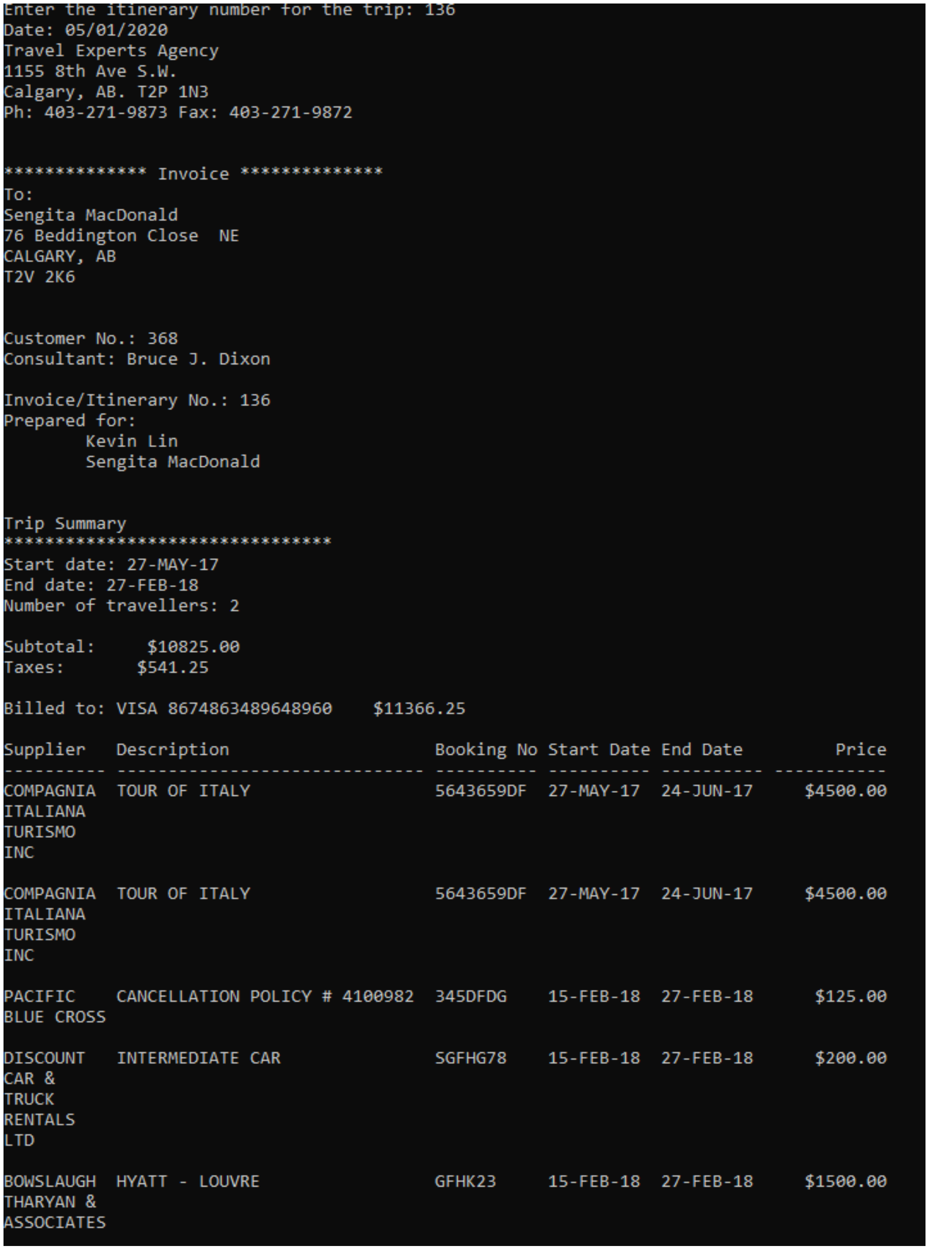
index creation script is updated *12-13, 23-24: Export Volatile data*	***13-14: Meeting with manager*** *12-13, 23-24: Export Volatile data*	*12-13, 23-24: Export Volatile data*	*12-13, 23-24: Export Volatile data*	*12-13, 23-24: Export Volatile data*	***14-15: Practice backup procedures in testing environment*** *12-13, 23-24: Export Volatile data*	*12-13, 23-24: Export Volatile data*
21 *1-3: Full Cold Backup* *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 23-24: Export Volatile data*	**22** *6-7: Export low volatile data* *12-13, 23-24: Export Volatile data 7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 23-24: Export Volatile data*	**23** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 23-24: Export Volatile data*	**24** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 23-24: Export Volatile data*	**25** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 23-24: Export Volatile data*	**26** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* ***14-15: Practice backup procedures in testing environment*** *12-13, 23-24: Export Volatile data*	**27** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 23-24: Export Volatile data*
28 *1-3: Full Cold Backup* *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 23-24: Export Volatile data*	**29** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 23-24: Export Volatile data*	**30** *6-7: Export low volatile data* *7-8: Backup alert log and clear* *8-9: Backup archive logs/Clear out the archive logs* *9-10: recreate the indices, make sure the index creation script is updated* *12-13, 23-24: Export Volatile data*	*Last day of the week, validate the backups in the daily tapes, and store them in the weekly tape. Set up for the next week*	*Last day of the month, validate the weekly tapes, store them in the monthly.* *Set up the weekly and daily tapes for reuse in the new month.*	*Last day of the year, validate the monthly tapes, store them in the yearly. Set up tapes for the new year… If needed clear out the oldest (7 years old) for the new backup.*	

- *Throughout each day (9-23): every 40-60 minutes do an online tablespace backup of the volatile tablespaces, only freezing one of them at a time*
- *Throughout each day (9-23): every 3 hours do an online tablespace backup of the non-volatile tablespaces, only freezing one of them at a time*
- *During the off-hours (23-9): do an online tablespace backup for the volatile tablespaces every 3 hours*
- *At 4 do online tablespace backup for the non-volatile tablespaces*
- *Every 4 hours during regular hours, a guaranteed restore point will be created; at the end of every day any restore points which are older than 4 days will be dropped.*
- *The catalog will be resynced at the end of every day*
- *Catalog should also be backed up.*

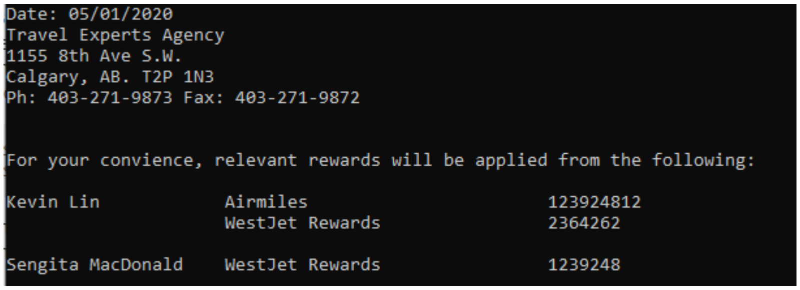
Sample Reports

The following are example reports that can be generated using the database.

Itinerary/Invoice

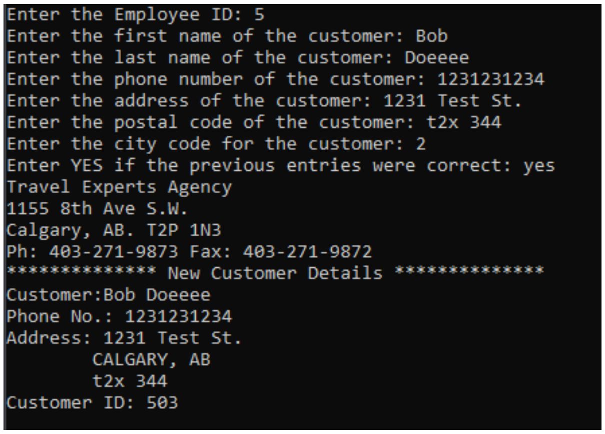


```
Enter the itinerary number for the trip: 136
Date: 05/01/2020
Travel Experts Agency
1155 8th Ave S.W.
Calgary, AB. T2P 1N3
Ph: 403-271-9873 Fax: 403-271-9872


************** Invoice **************
To:
Sengita MacDonald
76 Beddington Close  NE
CALGARY, AB
T2V 2K6


Customer No.: 368
Consultant: Bruce J. Dixon

Invoice/Itinerary No.: 136
Prepared for:
        Kevin Lin
        Sengita MacDonald


Trip Summary
********************************
Start date: 27-MAY-17
End date: 27-FEB-18
Number of travellers: 2

Subtotal:     $10825.00
Taxes:       $541.25

Billed to: VISA 8674863489648960    $11366.25

Supplier   Description                    Booking No Start Date End Date         Price
---------- ------------------------------ ---------- ---------- ---------- -----------
COMPAGNIA  TOUR OF ITALY                  5643659DF  27-MAY-17  24-JUN-17     $4500.00
ITALIANA
TURISMO
INC

COMPAGNIA  TOUR OF ITALY                  5643659DF  27-MAY-17  24-JUN-17     $4500.00
ITALIANA
TURISMO
INC

PACIFIC    CANCELLATION POLICY # 4100982  345DFDG    15-FEB-18  27-FEB-18      $125.00
BLUE CROSS

DISCOUNT   INTERMEDIATE CAR               SGFHG78    15-FEB-18  27-FEB-18      $200.00
CAR &
TRUCK
RENTALS
LTD

BOWSLAUGH  HYATT - LOUVRE                 GFHK23     15-FEB-18  27-FEB-18     $1500.00
THARYAN &
ASSOCIATES
```



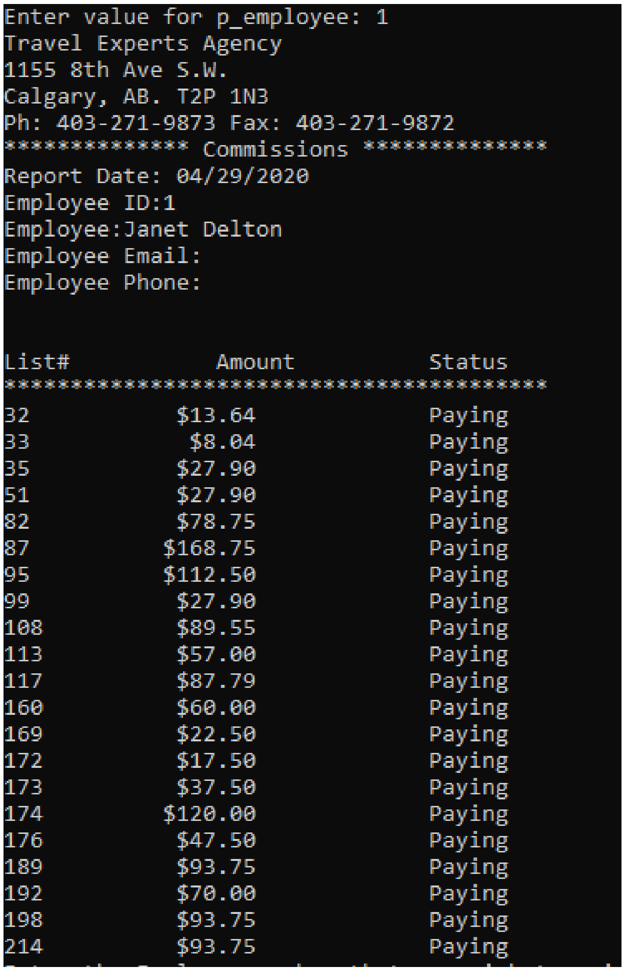
```
Date: 05/01/2020
Travel Experts Agency
1155 8th Ave S.W.
Calgary, AB. T2P 1N3
Ph: 403-271-9873 Fax: 403-271-9872


For your convience, relevant rewards will be applied from the following:

Kevin Lin           Airmiles                      123924812
                    WestJet Rewards               2364262

Sengita MacDonald   WestJet Rewards               1239248
```

New Customer Entry



```
Enter the Employee ID: 5
Enter the first name of the customer: Bob
Enter the last name of the customer: Doeeee
Enter the phone number of the customer: 1231231234
Enter the address of the customer: 1231 Test St.
Enter the postal code of the customer: t2x 344
Enter the city code for the customer: 2
Enter YES if the previous entries were correct: yes
Travel Experts Agency
1155 8th Ave S.W.
Calgary, AB. T2P 1N3
Ph: 403-271-9873 Fax: 403-271-9872
************** New Customer Details **************
Customer:Bob Doeeee
Phone No.: 1231231234
Address: 1231 Test St.
        CALGARY, AB
        t2x 344
Customer ID: 503
```

Commission



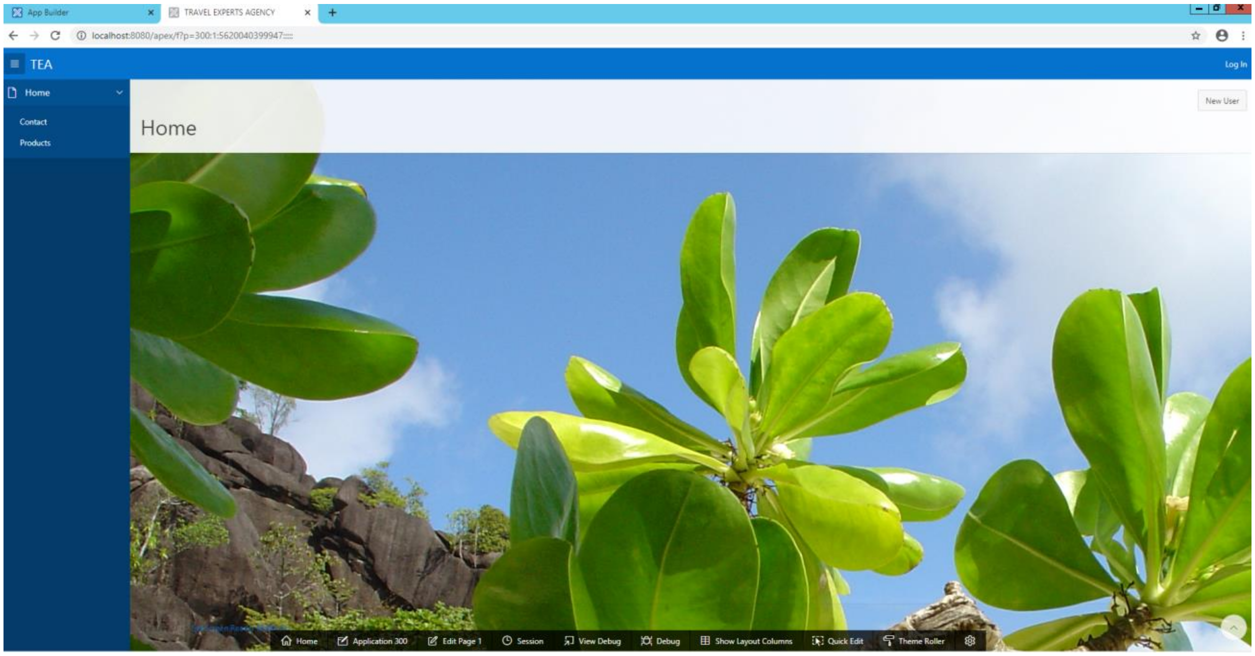
```
Enter value for p_employee: 1
Travel Experts Agency
1155 8th Ave S.W.
Calgary, AB. T2P 1N3
Ph: 403-271-9873 Fax: 403-271-9872
************** Commissions **************
Report Date: 04/29/2020
Employee ID:1
Employee:Janet Delton
Employee Email:
Employee Phone:


List#          Amount          Status
*****************************************
32            $13.64           Paying
33             $8.04           Paying
35            $27.90           Paying
51            $27.90           Paying
82            $78.75           Paying
87           $168.75           Paying
95           $112.50           Paying
99            $27.90           Paying
108           $89.55           Paying
113           $57.00           Paying
117           $87.79           Paying
160           $60.00           Paying
169           $22.50           Paying
172           $17.50           Paying
173           $37.50           Paying
174          $120.00           Paying
176           $47.50           Paying
189           $93.75           Paying
192           $70.00           Paying
198           $93.75           Paying
214           $93.75           Paying
```

Website – ApEx

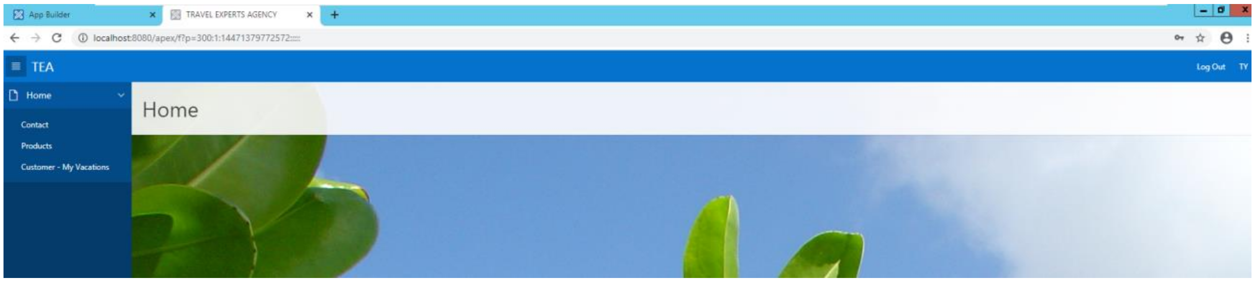
Public Pages

The Home Page

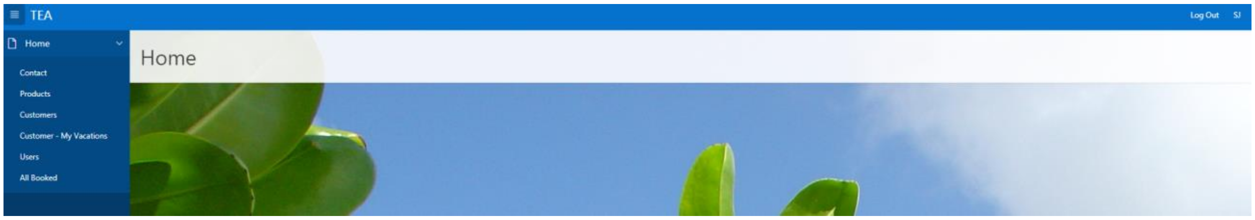
Allows for transition between it and the contact, products, log in and create new user.



When logged in, it also allows for more options. Such as the vacations that particular user has attached to their account.

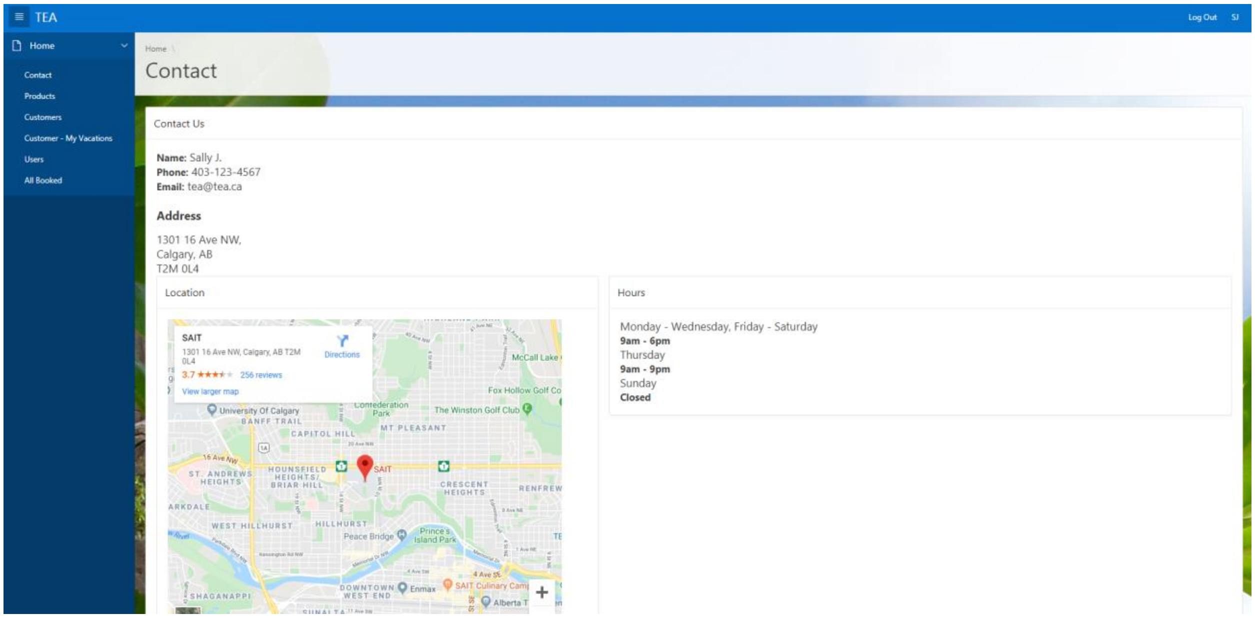


Or if admin, everything shows up…



Contact Page

Contains the agency's information with an interactable map.



Contact

Contact Us

Name: Sally J.
Phone: 403-123-4567
Email: tea@tea.ca

Address

1301 16 Ave NW,
Calgary, AB
T2M 0L4

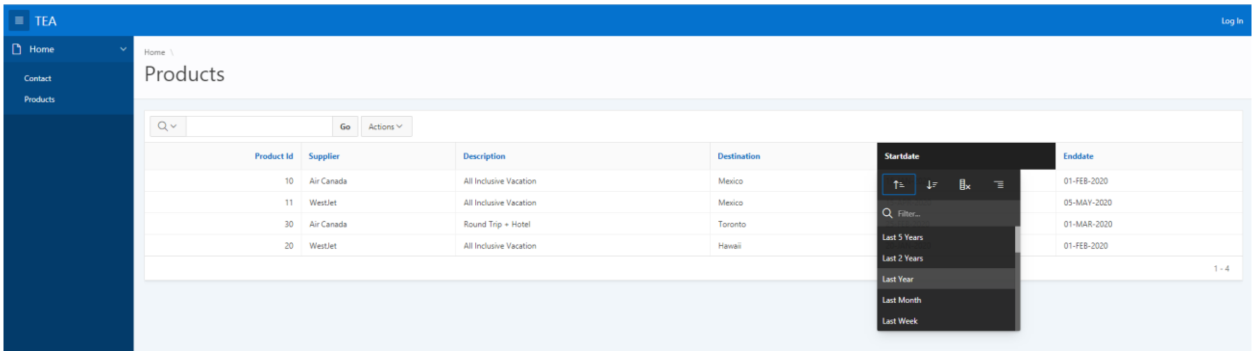
Location

Hours

Monday - Wednesday, Friday - Saturday
9am - 6pm
Thursday
9am - 9pm
Sunday
Closed

Vacation Packages

The Products page lists all the products that are available. As it is an interactive page, it can be sorted any way the user wishes.

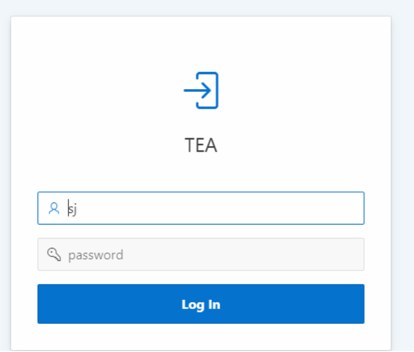


Products

Product Id	Supplier	Description	Destination	Startdate	Enddate
10	Air Canada	All Inclusive Vacation	Mexico		01-FEB-2020
11	WestJet	All Inclusive Vacation	Mexico		05-MAY-2020
30	Air Canada	Round Trip + Hotel	Toronto		01-MAR-2020
20	WestJet	All Inclusive Vacation	Hawaii		01-FEB-2020

Log In

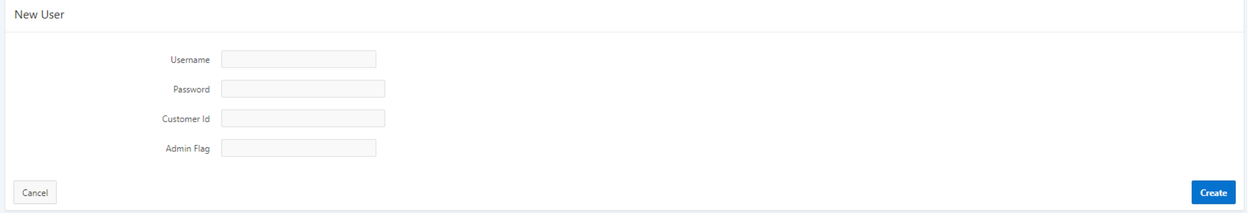
Standard log in page. Authenticates users using the users table.



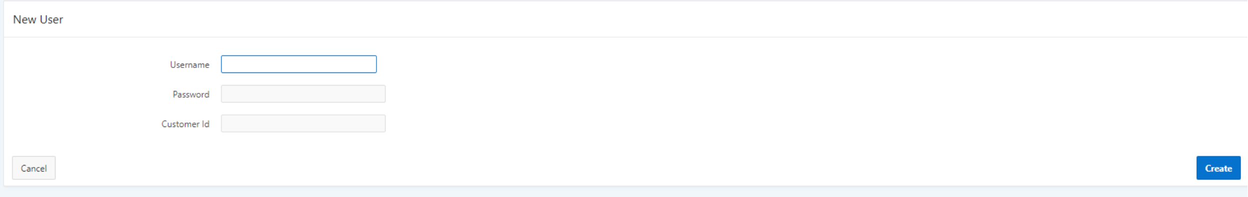
Create User

This page is only available if you are not logged in and on the home page, or through the users list that admins are able to access. If you cancel out, it will direct you straight to the products page. Only the admin will be able to set admin flags. Defaults to be ‘N’. It will not allow you to create a user with a username that already exists. Customer ID is an optional field for those customers that would like to see which vacation packages they have.

Admin View:



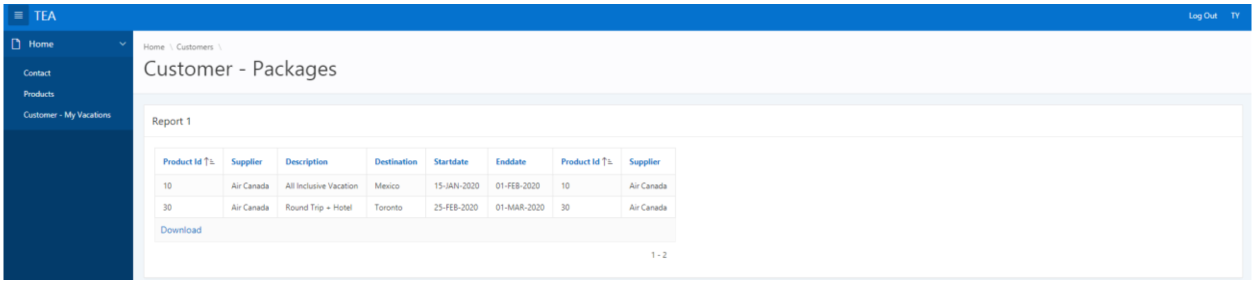
Public View:



Semi-Public Pages

Customer – My Vacations

This page reports all the products that the customer has bought.



Customer - Packages

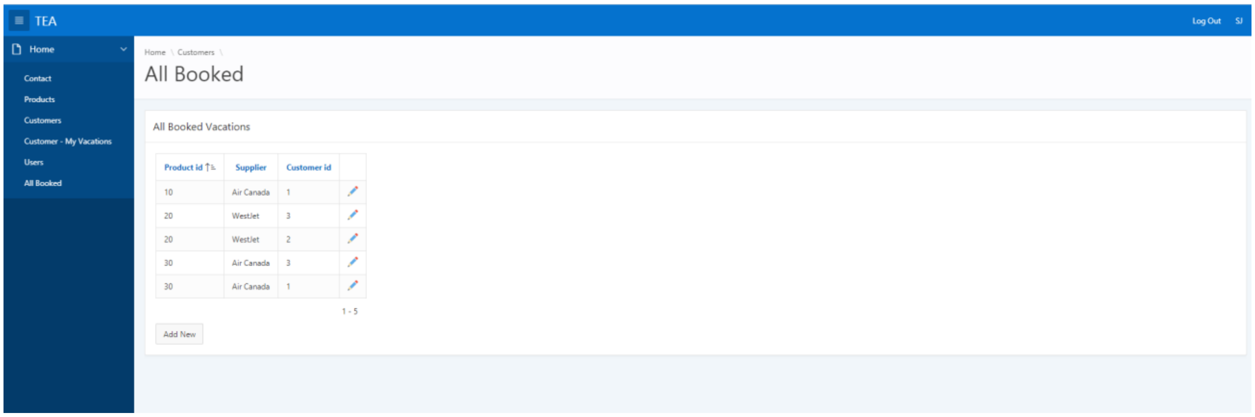
Report 1

Product Id ↑	Supplier	Description	Destination	Startdate	Enddate	Product Id ↑	Supplier
10	Air Canada	All Inclusive Vacation	Mexico	15-JAN-2020	01-FEB-2020	10	Air Canada
30	Air Canada	Round Trip + Hotel	Toronto	25-FEB-2020	01-MAR-2020	30	Air Canada

Private (Employee/Admin only) Pages

All Booked

This page lists all the products that have been bought by customers. There are also links in this page, which allows for edits, deletion or adding of new records.

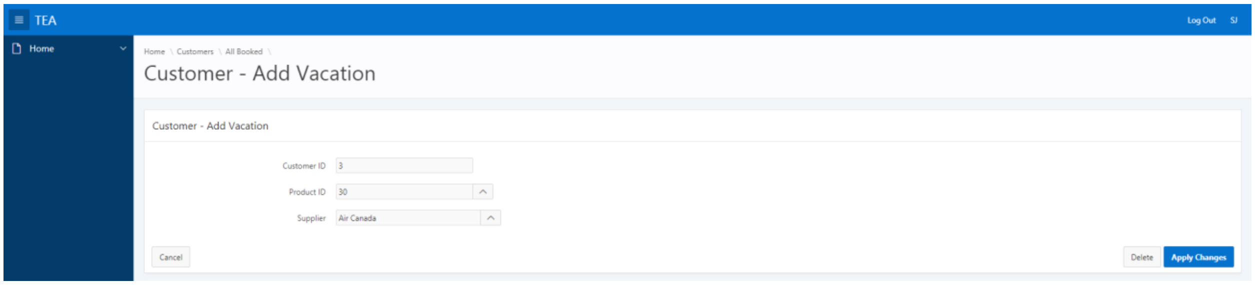


All Booked

All Booked Vacations

Product id ↑	Supplier	Customer id	
10	Air Canada	1	✎
20	WestJet	3	✎
20	WestJet	2	✎
30	Air Canada	3	✎
30	Air Canada	1	✎

1 - 5

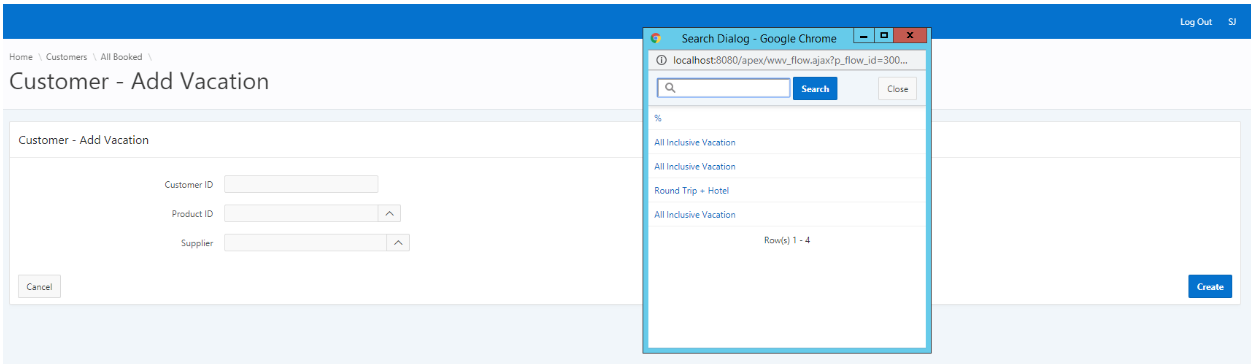


Customer - Add Vacation

Customer - Add Vacation

Customer ID	3
Product ID	30
Supplier	Air Canada

When adding, the product ID and suppliers can be searched for using the pop up list.


Log Out
SJ
Home \ Customers \ All Booked \

Customer - Add Vacation

Customer - Add Vacation

Customer ID	
Product ID	
Supplier	

Cancel
Create
Search Dialog - Google Chrome

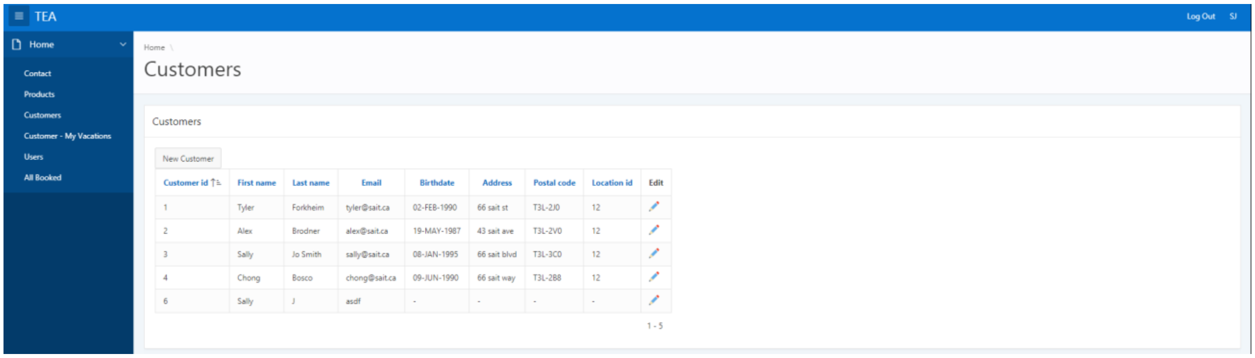
localhost:8080/apex/wwv_flow.ajax?p_flow_id=300...

%
All Inclusive Vacation
All Inclusive Vacation
Round Trip + Hotel
All Inclusive Vacation

Row(s) 1 - 4

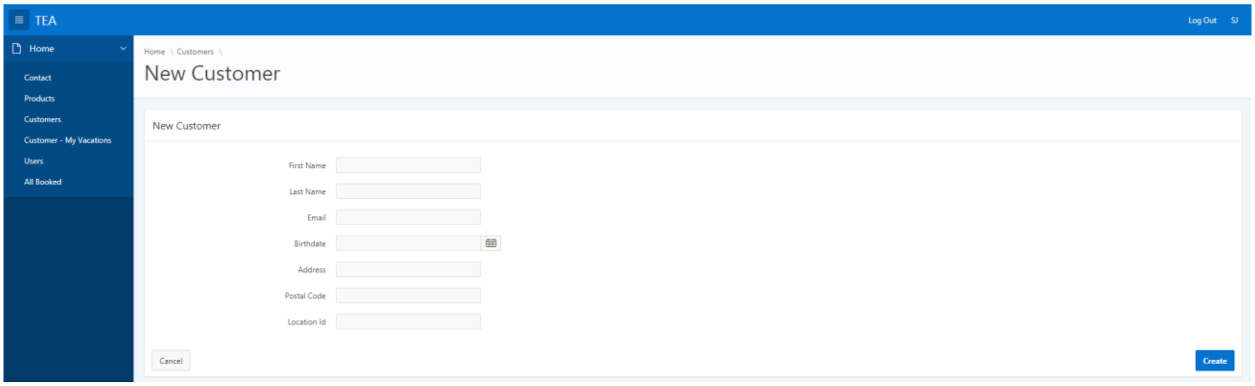
Customers

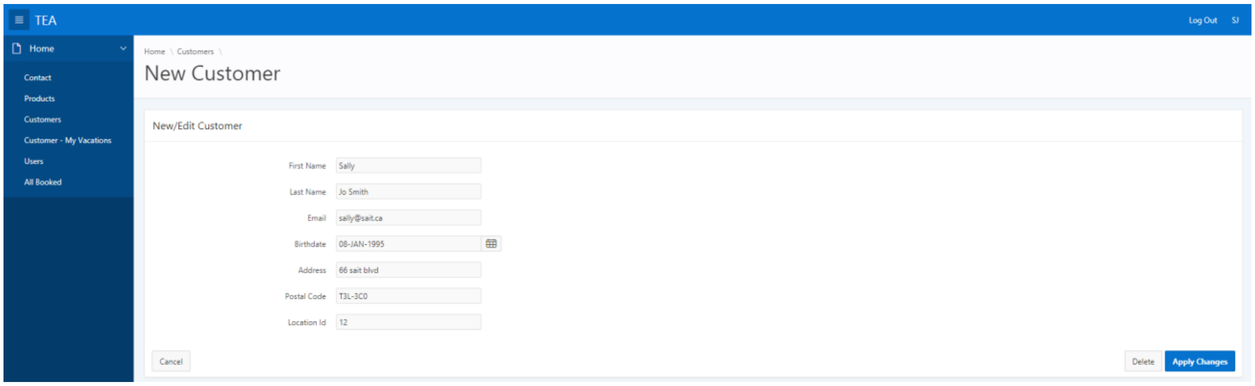
This page is used to keep track of all the customers, and their details. Links to edit/delete adding new customer page.



Customer id	First name	Last name	Email	Birthdate	Address	Postal code	Location id	Edit
1	Tyler	Forkheim	tyler@sait.ca	02-FEB-1990	66 sait st	T3L-2J0	12	✎
2	Alex	Brodner	alex@sait.ca	19-MAY-1987	43 sait ave	T3L-2V0	12	✎
3	Sally	Jo Smith	sally@sait.ca	08-JAN-1995	66 sait blvd	T3L-3C0	12	✎
4	Chong	Bosco	chong@sait.ca	09-JUN-1990	66 sait way	T3L-2B8	12	✎
6	Sally	J	asdf	-	-	-	-	✎

It will not allow you to add a new customer that has the exact same first and last name as an existing customer.

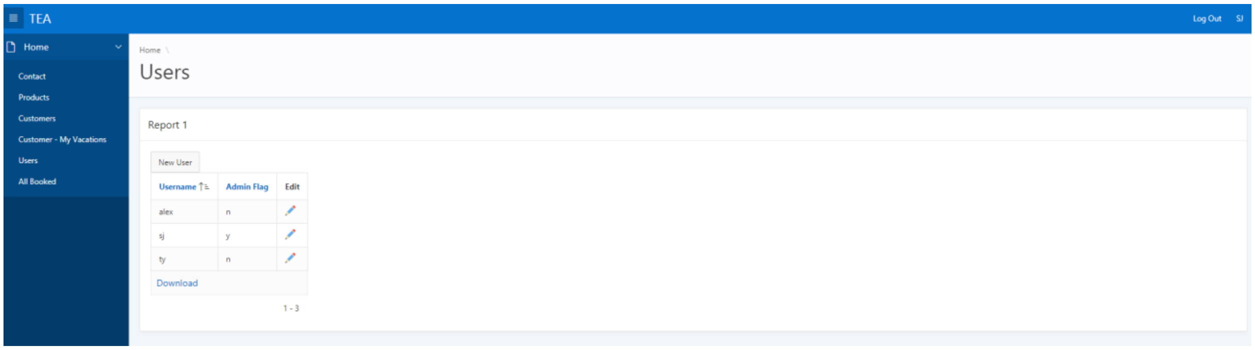




First Name	Sally
Last Name	Jo Smith
Email	sally@sait.ca
Birthdate	08-JAN-1995
Address	66 sait blvd
Postal Code	T3L-3C0
Location Id	12

Users

Lists all the current users that have a login credential. Allows for edits/deletes and adding.



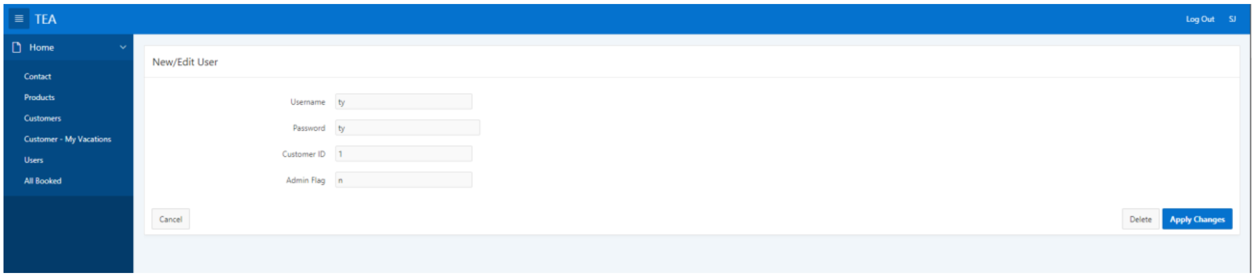
Users

Report 1

Username	Admin Flag	Edit
alex	n	✎
sj	y	✎
ty	n	✎

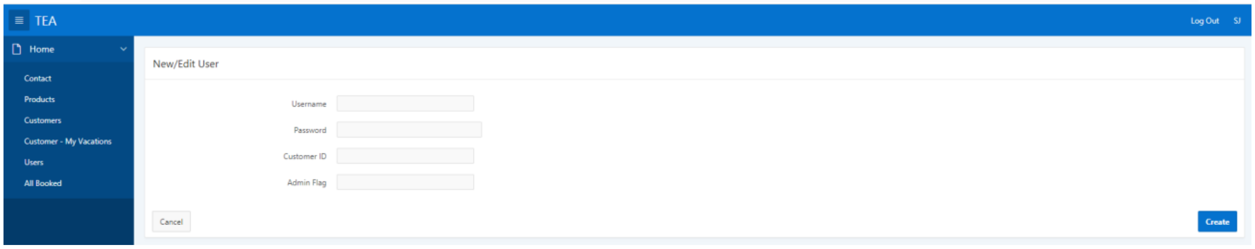
Download

1 - 3



New/Edit User

Username	ty
Password	ty
Customer ID	1
Admin Flag	n



New/Edit User

Username	
Password	
Customer ID	
Admin Flag	

Data Warehousing

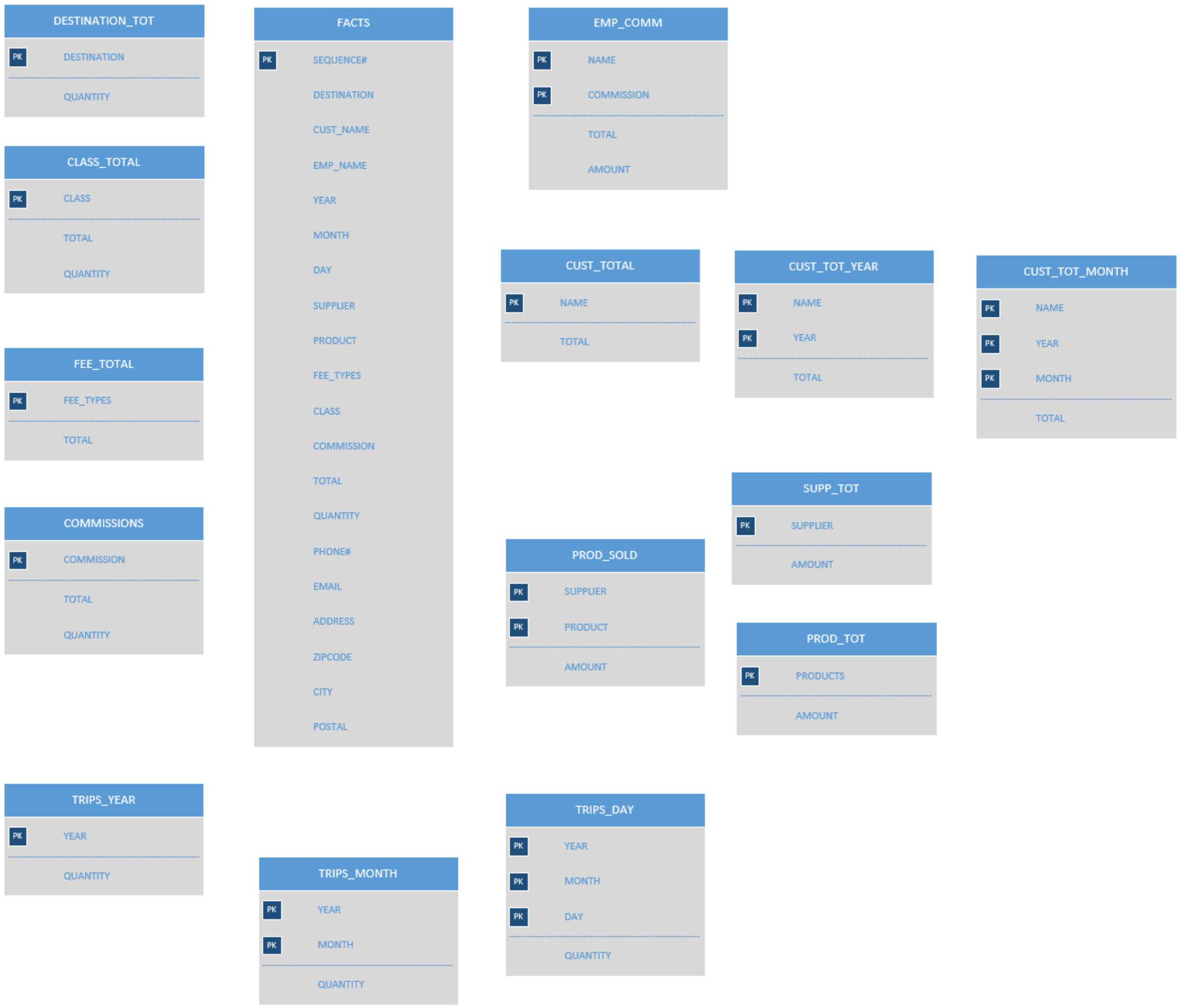
Marketing

This travel warehouse can keep track of employee commissions, customer sales, product sales, common destinations, total earnt from different fees, common trip times, and common trip classes. The commissions keep track of each employee's number of commissions in different categories – (booking, cancelled, outstanding, etc.). Using the GUI, many different reports can be generated. Simple reports using select all from the dimension tables can be used to generate reports as well.

Recommendation

If I had longer time to do this task, I would've reloaded the travel data using the UTL file, utilizing regular expressions to check for the data instead of the manual checks I did. I would also automate the updating of the dimensions table by using triggers instead of the merge commands. Alternatively I could've made them be scheduled procedures that can be called at specific time of day. Another thing that I would've done with more time is create more procedures for reports.

Warehouse - ERD



Appendix A – Workshops

Project Workshop - 1

Topic:

Define The Project.

Description:

The students will be introduced to a small travel agency that requires a database. Sample data and background information will be provided to the students.

The project team will develop questions for a senior travel agent, the manager, and the owner of Travel Experts travel agency. Each group will be able conduct an interview with the people mentioned above. The information collected from the interview together with the background information will form the basis for all work conducted on the Project in the following weeks.

Students may choose to develop their project schedule in Microsoft Project.

The students will be given an opportunity to work as a group to develop:

1. Requirements document
2. Project Schedule

Evaluation:

Each student will be assigned a grade for the workshop based on the contents of their Project Binder. Although students work in teams during the workshop class time, each team member is expected to produce their own binder materials. Students may not pool their resources and submit a single set of documents for the group.

Students will also be evaluated for team participation both by their peers and by the instructor.

Class Time Guidelines:

2 days

Project Binder Additions:

1. Interview Questions and Answers
2. Requirements Document
3. Preliminary Budget
4. Project Schedule
   Schedule must include:
   - Major tasks or modules to be completed
   - Detailed time-line for the project
   - Personnel to be assigned to the project

Project Workshop - 2

Topic:

Model the Database Schema

Description:

Students will apply the knowledge gained in Database concepts, SQL and Data Modelling to develop an appropriate model for the required project database. Students have the option of using Microsoft Visio for the development of their ER diagram.

Students must modify their Project Schedule to reflect the time and resources they feel are needed to complete a database model and related tasks.

The students will be given an opportunity to discuss their ideas with their Team Members during the workshop.

Evaluation:

Each student will be assigned a grade for the workshop based on the contents of their Project Binder. Although students work in teams during the workshop class time, each team member is expected to produce their own binder additions. Students may not pool their resources or submit a single set of documents for the group.

Students will also be evaluated for team participation both by their peers and by the instructor.

Class Time Guideline:

1 day

Project Binder Additions:

1. ER Diagram.
2. Documentation that fully describes all intended tables translated from your ER Diagram (entities/attributes/relationships).
3. Updated project tasks and schedule.

Project Workshop - 3

Topic:

Create the Travel Experts database.

Description:

Students will create the project database and necessary tablespaces with an appropriate disk layout.

Students should modify their Project Schedule to reflect the time and resources they feel are needed to configure and create a database.

Class Time Guideline:

1 day

Project Binder Additions:

1. Description of disk configuration used for database (“Blueprint”).
2. Steps to complete the create database procedure (“Recipe”).
3. Appropriate init.ora file for the database.
4. The Create commands used in scripts to create the database.
5. The following summary reports using the indicated views and other appropriate dictionary views:

a. Data files – v$datafile

b. Log files – v$logfile

c. Control files – v$controlfile

Tablespaces – dba_tablespaces…and other appropriate views.

Students are expected to format their reports.

6. Updated Project tasks and schedule

Project Workshop - 4

Topic:

Creation of Tables and Loading Data.

Description:

Students will create their tables and load their project data. Participants must create tables and constraints as described in their data model. Students are expected to create the following:

1. A set of tablespaces appropriate to the needs of their data model.
2. A user who will "own" the Travel Experts tables. This new user must have appropriate default and temporary tablespaces.

Students are allowed to modify their data model if they discover during the course of this Project Workshop that changes are necessary.

Students should modify their Project Schedule to reflect the time and resources they feel are needed to create tables and load data.

Class Time Guideline:

2 days…Students will continue working on this workshop during the time allotted to Project Workshop – 5 the following week.

Project Binder Additions:

1. Updated copy of the data model.
2. A set of scripts containing the SQL commands used to create all tables as indicated in data model. Commands used to create constraints must be included in the scripts. Tables must belong to an appropriate schema.
3. Output from describe commands confirming the structure of all tables.
4. Copy of SQL*Loader control files and log files for each data file loaded.
5. All scripts used to generate or insert data into tables that was not done through SQL*Loader.
6. Report on the number of records in each table after data has been loaded. Sample records (1-5 records) from each table. All reports must be appropriately formatted.
7. Updated reports from data dictionary tables and v$ views if structural changes are made to the database. Reports may include the following:
    a. Tablespaces and corresponding datafiles
    b. Tables created in the schema
    c. Database users
8. Updated Project tasks and schedule.

Project Workshop - 5

Topic:

Users, roles, profiles and security including Interim Update of Project.

Description:

Students will develop a set of roles and users for their project from the information gathered from interviews conducted with the Travel Agency. Groups of users representing the junior, intermediate and senior travel agents will be created in the database. In addition, students should create a manager and database administrator user accounts.

Students will implement resource and security profiles if they are applicable to their project. If profiles are used, they must be assigned to the appropriate users or groups of users.

Students will also implement auditing or other security measures if applicable to their project.

Students should modify their Project Schedule to reflect the time and resources they feel are needed to develop roles and profiles and implement security measures.

Class Time Guideline:

1 day

Project Binder Additions:

1. The following scripts:
    a. Create all users who must access the database.
    b. Create all roles and assign privileges to those roles
    c. Create profiles and assign to users.

2. The commands and/or steps required to establish auditing or other security measures.

3. The following reports and the commands needed to generate the reports:
    a. All database users/groups of users
    b. All created roles and the privileges assigned to each role.
    c. Description of users who are assigned each role.
    d. Resource and security profile limits and the users or groups of users assigned each profile.
4. Updated project tasks and schedule.

5. Present an update on the progress of this project to Travel Experts management/agents.
    i. Indicate the development and evolution of the requirements to date.
    ii. Specify how the timeline relates to actual completion of tasks and milestones.
    iii. Where do we go from here? What does the future hold?
    iv. Areas of concern that your group anticipates.

Project Workshop - 6

Topic:

Networking

Description:

Students will set up client-server access for their project databases.

A user account (for example: called AGENT with a password of agent28#) will be established on the project database. This account will represent an agent in the Travel Agency. Students should limit the access for this user to SELECT & INSERT only - on their database tables (so you they Insert the students name in a table and verify…possibly).

Students will test their connection within their group. Each group's tnsnames.ora file must allow for connections to 2 other databases. Although this configuration will not be part of the final configuration supplied to the Travel Agency, it will assist in testing.

Students should modify their Project Schedule to reflect the time and resources they feel are needed to successfully network a database.

Class Time Guideline:

1 day

Project Binder Additions:

All commands/scripts and steps for setting up the Travel Database Networking Scenario.

TNSNAMES.ORA file

– File must have address information for 2 databases other than the databases stored locally on the student's machine.

LISTENER.ORA file

- File must set up a listener for the project database stored locally on the student's machine.

SQLNET.ORA file

Updated project tasks and schedule.

Project Workshop - 7

Topic:

Data Warehouse & Data Mining

Description:

Students will create summary views or model a data warehouse for the travel agency. Metadata will be established for the data warehouse or summary views. Sample reports from the summary views or data warehouse will be generated. These reports will support marketing decision making in the travel agency

The students will generate marketing-related questions that might be answered through data mining activities. Recommendations concerning additional data the agency should collect will be made.

Students should modify their Project Schedule to reflect the time and resources they feel are needed to design and implement a data warehouse and complete simple data mining operations.

Class Time Guideline:

1-2 days

* students may be given time to work on this workshop during the scheduled data mining class time.

Project Binder Additions:

1. Recommended data warehouse model or scripts creating summary views
2. Metadata where required
3. Sample reports in support of marketing decision making
4. Marketing-related questions that might be answered through data mining activities.
5. Recommendations for future data collection
6. Updated project tasks and schedule.

Project Workshop - 8

Topic:

Backup and Recovery Strategies

Description:

Students will design an appropriate backup plan for Travel Experts Travel Agency. Students must consider the use of the Recovery Manager for the Agency (Project Day #2).

Students should also recommend, implement and test an appropriate backup plan for the working copy of their Travel Experts databases in the classroom. “Someone” could lose the whole or portion of their database at any time… if you get the hint??

The importance of regular backups of the Project Workshop database cannot be stressed enough. The loss of the project database may result in the inability of a student to complete the Project.

Students are encouraged to use the Recovery Manager for their test database backups.

Students should modify their Project Schedule to reflect the time and resources they feel are needed to design and implement an appropriate backup plan.

Class Time Guideline:

2 days

Project Binder Additions:

1. Description of a comprehensive backup plan/strategy for the travel agency.
2. Updated project tasks and schedule.

Project Workshop - 9

Topic:

Database testing, script building and reports

Description:

Students will generate a set of scripts intended to test their database structural design using the attached information. A sample invoice as well as other required reports identified in the Requirements document will be produced. The database ERD may be revised as necessary.

Students must generate their invoice, reports and data entry scripts using SQL*Plus, SQL and PL/SQL. No other products such as Oracle Forms and Reports or programming languages such as Visual Basic will be allowed.

Students should modify their Project Schedule to reflect the time and resources they feel are needed to develop the reports and scripts included in this workshop.

Class Time Guideline:

1 day

Project Binder Additions:

1. Sample invoice/itinerary similar to the examples provided by the Agency. Include the scripts used to generate these documents.
2. Different sample commission reports showing amount of commission earned with its due date; outstanding commissions; commissions paid… with scripts used to generate these reports.
3. One or more scripts that prompt for the entry of a new customer and transaction details. Include a sample of a new customer being added to the system.
4. Updated project tasks and schedule.

Project Workshop - 10

Topic:

Performance & Tuning

Description:

The students will make recommendations for optimal performance of their Travel database to the management of Travel Experts through inclusion in their Proposals. This will be submitted on the final day of the program. A separate document with the Performance & Tuning recommendations only should be included in your binder as well.

Students should modify their Project Schedule to reflect the time and resources they feel are needed to determine how performance may be optimized.

Class Time Guideline:

1 day

Project Binder Additions:

1. Recommendations for optimal performance
2. Updated project tasks and schedule.

Project Workshop - 11

Topic:

Database System Analysis and Proposal Writing.

Description:

Students will evaluate the most appropriate database system and operating system for their database. A proposal document outlining the complete project for the Travel Agency will be started.

A complete description of the required proposal document will be handed out to the students. Please review this document carefully as your proposal is worth approximately 20% of your project workshop assignment marks.

Class Time Guideline:

4 days

Project Binder Additions:

1. Report on best software/hardware solution for the travel agency with details on reasons for selecting/rejecting these components.
2. Proposal outlining details of project.

Project Workshop - 12

Topic:

Web integration – using Application Express (ApEx)

Description:

Students will consider the integration of their database with the web. A survey of travel-related websites is encouraged. Students will include their recommendations in their future project proposal.

To demonstrate how web technology can be used to provide access to the travel database, students will design and construct a web-based front-end to enhance the presentation and appearance of their database during their future Final Presentations. While the final presentation should have more features, the current assignment for this workshop will focus on some basic features (the framework or building blocks) as described below.

Class Time Guideline:

1 day Class time – Due: 1 week.

Project Binder Additions:

1. A discussion of how integration with the web would benefit or not benefit the agency.
2. Applicable, Hard-copy versions of sample websites.
3. Hard-copy of the sample web-pages that you have built during this workshop.

Deliverables:

1. Build a web page that will be the public entry page for your travel website. There should be some information about the agency, a list of links to other pages in your website, and any other features that you would like to include that will make your site comparable with other travel agencies. Your site should include:
2. a "Contact Us" page that has information about how to phone, email, or go to the agency
3. a vacation package page that will list from the database the details of all vacation packages available (expired packages should not appear, customers should be able to choose packages by date range).
4. a "Customer Entry" page that will be used only by the agent to enter customer data into the database when a customer places an order. (Include appropriate validation checks of the data).

Project Workshop - 13

Topic:

Oracle Hot Backup Shell Script and a datapump backup script

Description: hot backup 80%

Students will create an Oracle Hot Backup Shell script called *orahot.sh*. Participants must create a shell script that will accomplish the following:

Accept an instance name as the first parameter.

Use a variable to set a directory name for the destination location for the backup files.

Validate parameters and variables where applicable

Ensure database is running and is in ARCHIVELOG mode.

Back up all required database files using BEGIN TABLESPACE BACKUP and END TABLESPACE BACKUP commands.

Back up all required archived log files.

Back up the control file (both binary and logical).

Generate a runtime output log of all messages generated by the script execution.

Display start time of script

Display details of all files backed up

Display completion time of script

Generate a recovery script called *orahot_recovery_<instance_name>.sh*

Has commands to restore backup up files to their original locations

This script must be GENERATED by each run of the orahot.sh script.

Handle all necessary error conditions.

Description: Datapump 20%

Students will create an Oracle Datapump called *oradp.sh*. Participants must create a shell script that will accomplish the following:

Accept an instance name as the first parameter.

Use a variable to set a directory name for the destination location for the backup files.

Validate parameters and variables where applicable

Compresses the datapump backup on completion

Appendix B – Database Creation Steps

Creating new instance

1. In the Pfile (init[dbname]12c.ora) of the new database that you wish to create, be sure to specify password to be exclusive, and the instance name to be the new instance you wish to create.
2. Open up Command Prompt as the Administrator.
3. Change directory to the main drive. (Ex. cd C:\)
4. Set oracle_sid=[instance_name]
5. Run oradim: oradim -new -sid [instance_name] -syspwd [NewPassword]
   - NewPassword = This password should be at least 8 characters (max 128) and include lower & upper case letters, numbers and a special character – some special characters may require double quotes (“ ”) to encapsulate it
6. Verify & Confirm
   - Open Services and check for OracleService[InstanceName]
   - Check password file location (ex. C:\?\database\PWD[InstanceName].ora)
7. Using the same Command Prompt you set the oracle_sid for (or repeat steps 2-3 in a new command prompt), run sqlplus /nolog
8. Connect to an idle instance (ex. conn sys as sysdba)
9. Verify & Confirm Instance (using v$instance)

Creating new database

1. Prep the OS – Create format disks, create folders, etc.
2. Create a pfile (init[dbname]12c.ora), specifying the parameters needed.
3. Prepare an SQL file or two, one for database, one for tablespaces creation, using parameters needed
4. Connect using the proper instance, as sysdba
5. Verify & Confirm Instance (using v$instance)
6. Start up the new database in nomount (ex. startup nomount pfile = ‘[pfile_location \initTEA12c.ora]’
7. Run the create database script (Ex. SQL> c:\[filepath]\createDB.sql)
8. Run the create tablespace script if separate.
9. Verify the newly created database

Running CATALOG & CATPROC

DO NOT SPOOL THE CATALOG AND CATPROC FILES

1. Sql>connect sys as sysdba
2. Verify(using dba_registry)
3. Set termout, spool and echo off
4. Sql>@?\rdbms\admin\catalog.sql
   - ? Refers to oracle home directory.
5. Sql> @?\rdbms\admin\catproc.sql
6. Verify after running catproc (using dba_registry)
7. Since some of the results are Invalid we have to change the INVALID status to Valid status
   - Sql>@?\rdbms\admin\utlrp.sql

Running Pupbld

1. Connect as system, NOT SYS
2. Verify &Confirm (using the lack of product_privs)
3. sql> @?\sqlplus\admin\pupbld.sql
4. Verify &Confirm (using product_privs)

Create a new spfile for backup

1. Connect to sys as sysdba
2. Run Create spfile='[file_location\spfileTEA.ora]' from pfile='[pfile_location\initTEA.ora]';
3. Verify & Confirm
   - Check the folder Sp

Appendix C – TEA Table instance Charts

TABLE NAME: Affiliation

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
AFFILIATIONCODE	PK	NN/UNIQUE			CHAR	10	ACTA	ACTANEW	NEW
DESCRIPTION					VARCHAR2	100	Association of Canadian Travel Agents		

TABLE NAME: Supplier

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
SUPPLIERNO	PK	NN/UNIQUE			INT		12	30	37
SUPPLIERNAME					VARCHAR2	100	21st Century Travel	Disney cruise Line	Aer Lingus
AFFILIATIONCODE	FK		Affiliation	AFFILIATIONCODE	VARCHAR2	10	ACTA	PGY	ACTA
REPRESENTS	FK		Supplier	SUPPLIERNO	INT				12

TABLE NAME: SupplierDetail

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
SUPPLIERNO	PK/FK	NN	Supplier	SUPPLIERNO	INT		12	30	37
OFFICENO	PK	NN			NUMBER	3	1	1	1
CONTACTNAME		NN			VARCHAR2	50	Sean Russell	Linda Caldes	Franca Iuele
ADDRESS		NN			VARCHAR2	100	Suite 620 220 Duncan Mill Rd	Suite 1908 777 BaySt	Suite 400 160 Bloor St E
EMAIL					VARCHAR2	50	Email51@email.com	Email230@email.com	Email12312@email.com
PHONE					VARCHAR2	15	4164410166	4165993340	4169671510
FAX					VARCHAR2	20	4164411241	4165993405	4169677154
ZIP_POSTAL		NN			VARCHAR2	10	M3B 3J5	34747-4	M5W 1B9
CITYCODE		NN	City	CITYCODE	INT		1	2	1

TABLE NAME: TaxType

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
TAXTYPE	PK	NN/UNIQUE			NUMBER	3	1	2	3
TAXDESC		NN			VARCHAR2	100	GST	GST + PST	GST + PST + Environ. Tax

TABLE NAME: Product

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
PROUCTCATNO	PK/FK	NN	ProductCat	PRODUCTCATNO	NUMBER	3	100	600	300
SUPPLIERNO	PK/FK	NN	Supplier	SUPPLIERNO	INT		12	30	37
TAXRATE		NN			NUMBER	(3,2)	0.15	0.12	0.25
TAXTYPE	FK	NN	TaxType	TAXTYPE	NUMBER	3	1	2	3
PRICE		NN			NUMBER	(12,2)	90050.98	1358.12	54813.01

TABLE NAME: ProductCat

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
PRODUCTCATNO	PK	NN/UNIQUE			NUMBER	3	150	300	200
PRODUCTCATEGORY		NN			VARCHAR2	100	Airline	Cruise Lines	Attractions

TABLE NAME: Destination

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
DESTINATIONCODE	PK	NN/UNIQUE			VARCHAR2	5	ASIA	MED	AFR
DESTINDESC		NN			VARCHAR2	100	Asia	Mediterranean	Africa

TABLE NAME: Class

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
CLASSCODE	PK	NN/UNIQUE			VARCHAR2	5	FS	BSN	ECN
CLASSDESC		NN			VARCHAR2	100	FIRST CLASS	BUSINESS	ECONOMY

TABLE NAME: TripProduct

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
LISTNO	PK	NN/UNIQUE			INT		1	2	3
ITINERARYNUM	FK	NN	Trip	ITINERARYNUM	INT		1000	136	160
SUPPLIERNO	FK	NN	Supplier	SUPPLIERNO	INT		6346	842	3549
PRODUCTCATNO	FK	NN	ProductCat	PRODUCTCATNO	NUMBER	3	501	250	150
COMMISSIONAMOUNT		NN			NUMBER	(10,2)	130.11	200.11	12312.12

STARTDATE		NN			DATE		05-MAY-18	01-SEP-18	12-FEB-19
ENDDATE		NN			DATE		01-JUN-18	15-SEP-18	22-FEB-19
BOOKINGNUM		NN			VARCHAR2	15	56565T	SGFHG78	GHKE23
DESTINATIONCODE	FK		Destination	DESTINATIONCODE	VARCHAR2	5	EU	EU	AFR
CLASSCODE	FK		Class	CLASSCODE	VARCHAR2	5			DLX
DESCRIPTIONNO	FK	NN	Description	DESCRIPTIONNO	INT		1	2	3
COMMISSIONNO	FK	NN	CommissionType	COMMISSIONNO	NUMBER	2	1	2	1
EMPID	FK	NN	Employee	EMPID	INT		1	2	3
DUEDATE		NN			DATE		31-JUL-18	17-NOV-18	23-APR-19

TABLE NAME: CommissionType

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
COMMISSIONNO	PK	NN/UNIQUE			NUMBER	2	1	2	3
COMMISSIONTYPE		NN			VARCHAR2	25	PAID	OVERDUE	INVALID

TABLE NAME: TripTraveler

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
ITINERARYNUM	PK/FK	NN	Trip	ITINERARYNUM	INT		1000	136	160
TRAVELERNAME	PK	NN			VARCHAR2	50	ALEX	TYLER	AZIZ

TABLE NAME: Trip

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
ITINERARYNUM	PK	NN/UNIQUE			INT		1000	136	160
TRIPTYPECODE	FK	NN	TripType	TRIPTYPECODE	CHAR	2	B	L	G
FEECODE	FK	NN	FeeType	FEECODE	CHAR	3	BK	CH	RF
SALEDATE		NN			DATE		01-JAN-18	02-FEB-18	01-JAN-19
STARTDATE		NN			DATE		01-MAY-18	01-SEP-18	02-FEB-19
ENDDATE		NN			DATE		01-JUN-18	01-OCT-18	03-MAR-19
TRIPDESTINATION	FK		City	CITYCODE	INT		1	4	2
TRAVELERS		NN			NUMBER	3	4	2	1
CUSTOMERID	FK	NN	Customer	CUSTOMERID	INT		123	21	11

TABLE NAME: Description

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
DESCRIPTIONNO	PK	NN/UNIQUE			INT		1	2	3
DESCRIPTION		NN			VARCHAR2	100	All-Inc. European Tour	Safari Tour	Yacht Rental

TABLE NAME: TripType

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
TRIPTYPECODE	PK	NN/UNIQUE			CHAR	2	B	L	G
DESCRIPTION		NN/UNIQUE			VARCHAR2	50	BUSINESS	LEISURE	GROUP BOOKING

TABLE NAME: PaymentType

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
PAYMENTTYPE	PK	NN/UNIQUE			NUMBER	2	1	2	3
DESCRIPTION		NN/UNIQUE			VARCHAR2	50	DEPOSIT	FULL	FINAL PAYMENT

TABLE NAME: FeeType

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
FEECODE	PK	NN/UNIQUE			CHAR	3	RF	BK	CH
DESCRIPTION		NN/UNIQUE			VARCHAR2	50	Refund	Booking Charge	Change
FEEAMOUNT		NN			NUMBER	(5,2)	25	25	15

TABLE NAME: Employee

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
EMPID	PK	NN/UNIQUE			INT		1	2	3
FIRSTNAME		NN			VARCHAR2	15	Sally	Bandhavi	Abdul
LASTNAME		NN			VARCHAR2	20	Blue	Red	Green
HIREDATE		NN			DATE		01-SEP-19	02-SEP-19	03-SEP-19
PHONE					VARCHAR2	15	4032121245	4038748456	4038228555
EMAIL					VARCHAR2	50	aemail@email.com	Email2@email.com	Email3@email.com

TABLE NAME: Invoice

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
INVOICENUM	PK	NN			INT		1	2	3
PAYMENTTYPE	FK	NN	PaymentType	PAYMENTTYPE	NUMBER	2	1	1	2

ITINERARYNUM	PK/FK	NN	Trip	ITINERARYNUM	INT		441	123	412
BILLINGDATE		NN			DATE		15-APR-18	02-AUG-18	15-JAN-19
BILLAMOUNT		NN			NUMBER	(12,2)	500.25	1250.15	15500.99
REMAINDER		NN			NUMBER	(12,2)	1500	500	0

TABLE NAME: Customer

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
CUSTOMERID	PK	NN/UNIQUE			INT		123	21	11
EMPID	FK	NN	Employee	EMPID	INT		7	2	4
FIRSTNAME		NN			VARCHAR2	15	Ann	Bob	Charlie
LASTNAME		NN			VARCHAR2	20	Doe	Black	White
HPHONE					VARCHAR2	15	1231231234	3211231234	1324561324
BPHONE					VARCHAR2	15	6549874564	6547894654	1234567897
ADDRESS		NN			VARCHAR2	100	101 SOMEWHERE ST SE	10 OTHERPLACE CIR NW	123 SOMEPLACE ST NE

ZIP_POSTAL		NN			VARCHAR2	10	T2F 42D	T2X 3I1	T2A 4E2
CITYCODE		NN	City	CITYCODE	INT		1	2	1

TABLE NAME: OptionalCustDetail

Column Name	Key Type	Nulls/ Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
CUSTOMERID	PK/FK	NN/ UNIQUE	Customer	CUSTOMERID	INT		123	21	11
BIRTHDATE					DATE		29-MAY-90	11-NOV-84	12-SEP-70
WEDDING					DATE				01-JAN-98
EMAIL					VARCHAR2	50		1@email.com	
WHEELCHAIR					CHAR	1	Y		Y

TABLE NAME: CreditCard

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
CARDTYPENO	PK	NN/UNIQUE			NUMBER	2	1	2	3
DESCRIPTION		NN/UNIQUE			VARCHAR2	50	MASTERCARD	VISA	AMERICAN EXPRESS

TABLE NAME: Location

Column Name	Key Type	Nulls/Unique	FK Ref. Table	FK Ref. Col.	Data Type	Max.Length	Sample Data 1	Sample Data 2	Sample Data 3
CITYCODE	PK	NN/UNIQUE			INT		1	2	3
CITYNAME		NN			VARCHAR2	20	CALGARY	EDMONTON	VULCAN
STATEPROV	FK				Varchar2	5	AB	AB	AB
COUNTRY	FK	NN			Varchar2	20	Canada	Canada	Canada

TABLE NAME: CustCredit

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
CUSTOMERID	PK/FK	NN/UNIQUE			INT		123	21	11
CREDITCARDNO		NN			VARCHAR2	20	1234 1235 1345 4561	1234 4564 5554 1235	1234 5454 6565 8484
EXPDATE		NN			DATE		JAN-22	FEB-23	JAN-20
CARDTYPENO	FK	NN	CreditCard	CARDTYPENO	NUMBER	2	1	2	3

TABLE NAME: CustRewardCard

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
CARDNO	PK	NN			VARCHAR2	20	1224561231	1231231231	123124125
CARDTYPE	PK/FK	NN	RewardCard	CARDTYPE	NUMBER	3	1	2	3
CUSTOMERID	FK	NN	Customer	CUSTOMERID	INT		123	21	11

TABLE NAME: RewardCard

Column Name	Key Type	Nulls/Unique	FK Ref Table	FK Ref. Col.	Data Type	Max. Length	Sample Data 1	Sample Data 2	Sample Data 3
CARDTYPE	PK	NN/UNIQUE			INT		1	2	3
DESCRIPTION		UNIQUE			VARCHAR2	50	AIRMILE	WESTJET REWARDS	SOMEOTHER REWARDS CARD

Appendix D – Travel Agency Database Scripts

Adhoc Table Script

```
DROP TABLE DONOTTOUCH.cust_adhoc CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.supp_adhoc CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.sale_adhoc CASCADE CONSTRAINTS purge;


CREATE TABLE DONOTTOUCH.cust_adhoc
      (
      cust_id int
      ,first_name varchar2(15)
      ,last_name varchar2(20)
      ,agent varchar2(3)
      ,email varchar2(50)
      ,home_phone varchar2(15)
      ,business_phone varchar2(15)
      ,birthdate date
      ,address varchar2(100)
      ,city varchar2(20)
      ,postal varchar2(7)
      ,province varchar2(20)
      ,country varchar2(20)
      )
      TABLESPACE ADHOC;

CREATE TABLE DONOTTOUCH.supp_adhoc
      (
      prod_supp int
      ,prod_cat int
      ,supp_off int
      ,prod_desc varchar2(100)
      ,cont_name varchar2(50)
      ,company varchar2(100)
      ,address varchar2(50)
      ,address2 varchar2(50)
      ,city varchar2(50)
      ,prov varchar2(50)
      ,postal varchar2(10)
      ,country varchar2(20)
      ,phone varchar2(15)
      ,fax varchar2(15)
      ,email varchar2(50)
      ,web varchar2(50)
      ,represents varchar2(50)
      ,affiliation varchar2(10)
      )
      TABLESPACE ADHOC;

CREATE TABLE DONOTTOUCH.sale_adhoc
      (
      sales_date date
      ,cust_id int
      ,itinerary varchar2(20)
      ,agent varchar2(3)
      ,booking varchar2(20)
      ,prod_cat int
      ,supplier_id int
      ,supp_off int
      ,trip_start date
      ,trip_end date
      ,class varchar2(10)
      ,num_travellers int
      ,product varchar2(20)
      ,description varchar2(100)
      ,destination varchar2(30)
      ,dest_id varchar2(10)
      ,credit_card varchar2(20)
      ,expiry date
      ,card_num varchar2(20)
      ,bill_date date
      ,bill_desc varchar2(50)
      ,base_price number
      ,tot_price number
      ,bill_amt number
      ,agency_fee_code varchar2(10)
      ,agency_fee_amt number
      ,agency_comm number
      )
      TABLESPACE ADHOC;

insert into donottouch.triptraveller
(select unique(a.itinerary), c.firstname||' '||c.lastname
from donottouch.sale_adhoc a join
donottouch.customer c on c.customerid=a.cust_id);
```

Create Table Script

```
/*
The following was used to create the user to hold all tables...
CREATE USER DONOTTOUCH IDENTIFIED BY oracle
DEFAULT TABLESPACE userdata01
TEMPORARY TABLESPACE temp01
QUOTA 500m ON userdata01
QUOTA 500m ON userdata02
QUOTA 300m ON indx
QUOTA 300m ON ADHOC
PROFILE dba
ACCOUNT LOCK;
*/
--have to be user donottouch to run this script
DROP TABLE DONOTTOUCH.AFFILIATION CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.CLASS CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.COMMISSION CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.COMMISSIONTYPE CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.CREDITCARD CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.CUSTOMER CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.CUSTREWARDCARD CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.CUSTCREDIT CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.DESTINATION CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.DESCRIPTION CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.EMPLOYEE CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.FEETYPE CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.INVOICE CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.LOCATION CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.OPTIONALCUSTDETAIL CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.PAYMENTTYPE CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.PRODUCT CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.PRODUCTCAT CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.REWARDCARD CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.SUPPLIER CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.TAXTYPE CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.TRIPPRODUCT CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.TRIPTYPE CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.TRIP CASCADE CONSTRAINTS purge;
DROP TABLE DONOTTOUCH.TRIPTRAVELLER CASCADE CONSTRAINTS purge;

DROP SEQUENCE donottouch.empidcount;
DROP SEQUENCE donottouch.paymenttypeseq;
DROP SEQUENCE donottouch.taxtypeseq;
DROP SEQUENCE donottouch.tripprodseq;
DROP SEQUENCE donottouch.locationseq;
DROP SEQUENCE donottouch.descseq;
DROP SEQUENCE donottouch.invoiceseq;
DROP SEQUENCE donottouch.tripseq;
DROP SEQUENCE donottouch.creditseq;
DROP SEQUENCE donottouch.custiddeq;
DROP SEQUENCE donottouch.rewardcardseq;


CREATE SEQUENCE donottouch.rewardcardseq
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;
CREATE SEQUENCE donottouch.empidcount
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;
CREATE SEQUENCE donottouch.taxtypeseq
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;
```

```
CREATE SEQUENCE donottouch.tripprodseq
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;
CREATE SEQUENCE donottouch.paymenttypeseq
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;
CREATE SEQUENCE donottouch.locationseq
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;
CREATE SEQUENCE donottouch.descseq
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;
CREATE SEQUENCE donottouch.invoiceseq
      INCREMENT BY 1
      START WITH 1
      NOCACHE;
CREATE SEQUENCE donottouch.tripseq
      INCREMENT BY 1
      START WITH 1135
      NOCYCLE
      NOCACHE;
CREATE SEQUENCE donottouch.creditseq
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;
CREATE SEQUENCE donottouch.custiddeq
      INCREMENT BY 1
      START WITH 500
      NOCYCLE
      NOCACHE;

CREATE TABLE DONOTTOUCH.PAYMENTTYPE
       (PAYMENTTYPE NUMBER(2) DEFAULT donottouch.paymenttypeseq.nextval
CONSTRAINT PAYMENTTYPE_PK PRIMARY KEY
           USING INDEX (CREATE INDEX PAYMENTTYPE_INDX ON
DONOTTOUCH.PAYMENTTYPE(PAYMENTTYPE)
           TABLESPACE INDX
           STORAGE(INITIAL 20K NEXT 20K PCTINCREASE 10))
      ,PAYMENTDESC VARCHAR2(50) CONSTRAINT PAYMENTTYPE_DESC_NN NOT
NULL)
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);
```

```
CREATE TABLE DONOTTOUCH.AFFILIATION
      (AFFILIATIONCODE VARCHAR2(10) CONSTRAINT AFFILIATION_PK PRIMARY
KEY
            USING INDEX (CREATE INDEX AFFILIATION_INDX ON
DONOTTOUCH.AFFILIATION(AFFILIATIONCODE)
            TABLESPACE INDX
            STORAGE (INITIAL 20K NEXT 20K PCTINCREASE 10))
      ,AFFILIATIONDESC VARCHAR2(100))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);


CREATE TABLE DONOTTOUCH.LOCATION
      (CITYCODE INT DEFAULT donottouch.locationseq.nextval CONSTRAINT
LOCATION_PK PRIMARY KEY
            USING INDEX (CREATE INDEX LOCATION_INDX ON
DONOTTOUCH.LOCATION(CITYCODE)
            TABLESPACE INDX
            STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,CITYNAME VARCHAR2(50) CONSTRAINT LOCATION_CITY_NN NOT NULL
      ,STATEPROV VARCHAR2(20)
      ,COUNTRYNAME VARCHAR2(20) CONSTRAINT LOCATION_COUNTRY_NN NOT
NULL)
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);


CREATE TABLE DONOTTOUCH.SUPPLIER
      (SUPPLIERNO INT
      ,COMPANYNAME VARCHAR2(100)
      ,AFFILIATIONCODE VARCHAR2(10) CONSTRAINT SUPPLIER_AFFILIATION_FK
REFERENCES DONOTTOUCH.AFFILIATION(AFFILIATIONCODE)
      ,REP_NO INT
      ,REPRESENTS VARCHAR2(100)
      ,OFFICENO NUMBER(3)
      ,CONTACTNAME VARCHAR2(50)
      ,ADDRESS VARCHAR2(100)
      ,EMAIL VARCHAR2(50)
      ,PHONE VARCHAR2(15)
      ,FAX VARCHAR2(15)
      ,WEB VARCHAR2(50)
      ,ZIP_POSTAL VARCHAR2(10)
      ,CITYCODE INT CONSTRAINT SUPPLIER_CITY_FK REFERENCES
DONOTTOUCH.LOCATION(CITYCODE)
      ,CONSTRAINT SUPPLIERDETAIL_PK PRIMARY KEY (SUPPLIERNO,
COMPANYNAME)
            USING INDEX (CREATE INDEX SUPPLIER_INDX ON
DONOTTOUCH.SUPPLIER(SUPPLIERNO, COMPANYNAME)
            TABLESPACE INDX
            STORAGE(INITIAL 20K NEXT 20K PCTINCREASE 10))
      ,CONSTRAINT SUPP_REP_FK FOREIGN KEY (REPRESENTS, REP_NO)
REFERENCES DONOTTOUCH.SUPPLIER(COMPANYNAME, SUPPLIERNO))
```

```
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);


CREATE TABLE DONOTTOUCH.TAXTYPE
      (TAXTYPE NUMBER(3) DEFAULT donottouch.taxtypeseq.nextval
CONSTRAINT TAXTYPE_PK PRIMARY KEY
           USING INDEX (CREATE INDEX TAXTYPE_INDX ON
DONOTTOUCH.TAXTYPE(TAXTYPE)
           TABLESPACE INDX
           STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,TAXDESC VARCHAR2(100) CONSTRAINT TAXTYPE_DESC_NN NOT NULL)
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);

CREATE TABLE DONOTTOUCH.PRODUCTCAT
      (PRODUCTCATNO NUMBER(3) CONSTRAINT PRODUCTCAT_PK PRIMARY KEY
           USING INDEX (CREATE INDEX PRODUCTCAT_INDX ON
DONOTTOUCH.PRODUCTCAT(PRODUCTCATNO)
           TABLESPACE INDX
           STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,PRODUCTCATEGORY VARCHAR2(100))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 40K NEXT 5K PCTINCREASE 10);

CREATE TABLE DONOTTOUCH.DESTINATION
      (DESTINATIONCODE VARCHAR2(5) CONSTRAINT DESTINATION_PK PRIMARY
KEY
           USING INDEX (CREATE INDEX DESTINATION_INDX ON
DONOTTOUCH.DESTINATION(DESTINATIONCODE)
           TABLESPACE INDX
           STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,DESTINDESC VARCHAR2(100) CONSTRAINT DESTINATION_DESC_NN NOT
NULL)
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);

CREATE TABLE DONOTTOUCH.CLASS
      (CLASSCODE VARCHAR2(5) CONSTRAINT CLASS_PK PRIMARY KEY
           USING INDEX (CREATE INDEX CLASS_INDX ON
DONOTTOUCH.CLASS(CLASSCODE)
           TABLESPACE INDX
           STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,CLASSDESC VARCHAR2(100) CONSTRAINT CLASS_DESC_NN NOT NULL)
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);

CREATE TABLE DONOTTOUCH.COMMISSIONTYPE
      (COMMISSIONNO NUMBER(2) CONSTRAINT COMMISSIONTYPE_PK PRIMARY KEY
           USING INDEX (CREATE INDEX COMMISSIONTYPE_INDX ON
DONOTTOUCH.COMMISSIONTYPE(COMMISSIONNO)
           TABLESPACE INDX
```

```
            STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,COMMISSIONTYPE VARCHAR2(25) CONSTRAINT COMMISSIONTYPE_DESC_NN
NOT NULL)
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);



CREATE TABLE DONOTTOUCH.DESCRIPTION
      (DESCRIPTIONNO INT DEFAULT donottouch.descseq.nextval CONSTRAINT
DESCRIPTION_PK PRIMARY KEY
            USING INDEX (CREATE INDEX DESCRIPTION_INDX ON
DONOTTOUCH.DESCRIPTION(DESCRIPTIONNO)
            TABLESPACE INDX
            STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,DESCRIPTION VARCHAR2(100) CONSTRAINT DESCRIPTION_NN NOT NULL)
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);


CREATE TABLE DONOTTOUCH.TRIPTYPE
      (TRIPTYPECODE CHAR(2) CONSTRAINT TRIPTYPE_PK PRIMARY KEY
            USING INDEX (CREATE INDEX TRIPTYPE_INDX ON
DONOTTOUCH.TRIPTYPE(TRIPTYPECODE)
            TABLESPACE INDX
            STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,TRIPDESC VARCHAR2(50) CONSTRAINT TRIPTYPE_DESC_NN NOT NULL)
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);


CREATE TABLE DONOTTOUCH.EMPLOYEE
      (EMPID INT DEFAULT donottouch.empidcount.nextval CONSTRAINT
EMPLOYEE_ID_PK PRIMARY KEY
            USING INDEX (CREATE INDEX EMPLOYEE_INDX ON
DONOTTOUCH.EMPLOYEE(EMPID)
            TABLESPACE INDX
            STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,FIRSTNAME VARCHAR2(15) CONSTRAINT EMPLOYEE_FNAME_NN NOT NULL
      ,LASTNAME VARCHAR2(20) CONSTRAINT EMPLOYEE_LNAME_NN NOT NULL
      ,HIREDATE DATE DEFAULT sysdate CONSTRAINT EMPLOYEE_HIRE_NN NOT
NULL
      ,PHONE VARCHAR2(15)
      ,EMAIL VARCHAR2(50))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);

CREATE TABLE DONOTTOUCH.FEETYPE
      (FEECODE CHAR(3) CONSTRAINT FEECODE_PK PRIMARY KEY
            USING INDEX (CREATE INDEX FEECODE_INDX ON
DONOTTOUCH.FEETYPE(FEECODE)
            TABLESPACE INDX
            STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,FEETYPEDESC VARCHAR2(50) CONSTRAINT FEETYPE_DESC_NN NOT NULL
      ,FEEAMOUNT NUMBER(5,2))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);


CREATE TABLE DONOTTOUCH.CUSTOMER
      (CUSTOMERID INT DEFAULT custiddeq.nextval CONSTRAINT CUSTOMER_PK
PRIMARY KEY
            USING INDEX (CREATE INDEX CUSTOMER_INDX ON
DONOTTOUCH.CUSTOMER(CUSTOMERID)
            TABLESPACE INDX
            STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,EMPID INT CONSTRAINT CUST_EMP_FK REFERENCES
DONOTTOUCH.EMPLOYEE(EMPID)
      ,FIRSTNAME VARCHAR2(15) CONSTRAINT CUSTOMER_FNAME_NN NOT NULL
      ,LASTNAME VARCHAR2(20) CONSTRAINT CUSTOMER_LNAME_NN NOT NULL
      ,HPHONE VARCHAR2(15)
      ,BPHONE VARCHAR2(15)
      ,ADDRESS VARCHAR2(100) CONSTRAINT CUST_ADDRESS_NN NOT NULL
      ,CITYCODE INT CONSTRAINT CUST_CITY_FK REFERENCES
DONOTTOUCH.LOCATION(CITYCODE)
      ,ZIP_POSTAL VARCHAR2(10) CONSTRAINT CUST_ZIP_NN NOT NULL)
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);

CREATE TABLE DONOTTOUCH.REWARDCARD
      (CARDTYPE NUMBER(3) DEFAULT rewardcardseq.nextval CONSTRAINT
REWARDCARD_PK PRIMARY KEY
            USING INDEX (CREATE INDEX REWARDCARD_INDX ON
DONOTTOUCH.REWARDCARD(CARDTYPE)
            TABLESPACE INDX
            STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,REWARDCARDDESC VARCHAR2(50) CONSTRAINT REWARDCARD_DESC_NN NOT
NULL)
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);

CREATE TABLE DONOTTOUCH.CUSTREWARDCARD
      (CARDNO VARCHAR2(20)
      ,CARDTYPE NUMBER(3) CONSTRAINT CUSTREWARD_TYPE_FK REFERENCES
DONOTTOUCH.REWARDCARD(CARDTYPE)
      ,CUSTOMERID INT CONSTRAINT CUSTREWARD_CUST_FK    REFERENCES
DONOTTOUCH.CUSTOMER(CUSTOMERID)
      ,CONSTRAINT CUSTREWARDCARD_PK PRIMARY KEY (CARDNO, CARDTYPE)
            USING INDEX (CREATE INDEX CUSTREWARDCARD_INDX ON
DONOTTOUCH.CUSTREWARDCARD(CARDNO, CARDTYPE)
            TABLESPACE INDX
            STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10)))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);
```

```
CREATE TABLE DONOTTOUCH.CREDITCARD
      (CARDTYPENO NUMBER(2) DEFAULT donottouch.creditseq.nextval
CONSTRAINT CREDITCARD_PK PRIMARY KEY
           USING INDEX (CREATE INDEX CREDITCARD_INDX ON
DONOTTOUCH.CREDITCARD(CARDTYPENO)
           TABLESPACE INDX
           STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,CREDITDESC VARCHAR2(50) CONSTRAINT CREDITCARD_DESC_NN NOT NULL )
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);


CREATE TABLE DONOTTOUCH.CUSTCREDIT
      (CUSTOMERID INT CONSTRAINT CUSTCREDIT_ID_FK REFERENCES
DONOTTOUCH.CUSTOMER(CUSTOMERID)
      ,CREDITCARDNO VARCHAR2(20)
      ,EXPDATE DATE
      ,CARDTYPENO NUMBER(2) CONSTRAINT CUSTCREDIT_TYPE_FK REFERENCES
DONOTTOUCH.CREDITCARD(CARDTYPENO)
      ,CONSTRAINT CUSTCREDIT_PK PRIMARY KEY (CUSTOMERID, CREDITCARDNO)
           USING INDEX (CREATE INDEX CUSTCREDIT_INDX ON
DONOTTOUCH.CUSTCREDIT(CUSTOMERID, CREDITCARDNO)
           TABLESPACE INDX
           STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10)))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);


CREATE TABLE DONOTTOUCH.TRIP
      (ITINERARYNUM INT DEFAULT donottouch.tripseq.nextval
      ,TRIPTYPECODE CHAR(2) CONSTRAINT TRIP_TYPE_FK REFERENCES
DONOTTOUCH.TRIPTYPE(TRIPTYPECODE)
      ,CUSTOMERID INT CONSTRAINT TRIP_CUST_FK REFERENCES
DONOTTOUCH.CUSTOMER(CUSTOMERID)
      ,STARTDATE DATE
      ,ENDDATE DATE
      ,TRAVELLERS NUMBER(3)
      ,CONSTRAINT TRIP_PK PRIMARY KEY (itinerarynum)
           USING INDEX (CREATE INDEX TRIP_INDX ON
DONOTTOUCH.TRIP(ITINERARYNUM)
           TABLESPACE INDX
           STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10)))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);

CREATE TABLE DONOTTOUCH.INVOICE
      (INVOICENO INT DEFAULT donottouch.invoiceseq.nextval
      ,ITINERARYNUM INT CONSTRAINT INVOICE_ITINERARY_FK REFERENCES
DONOTTOUCH.TRIP(ITINERARYNUM)
```

```
      ,PAYMENTTYPE NUMBER(2) CONSTRAINT INVOICE_PAYMENT_FK REFERENCES
DONOTTOUCH.PAYMENTTYPE(PAYMENTTYPE)
      ,BILLINGDATE DATE CONSTRAINT INVOICE_DATE_NN NOT NULL
      ,BILLAMOUNT NUMBER(12,2)
      ,REMAINDER NUMBER(12,2)
      ,SALEDATE DATE DEFAULT sysdate CONSTRAINT TRIP_SALE_NN NOT NULL
      ,FEECODE CHAR(3) CONSTRAINT TRIP_FEECODE_FK REFERENCES
DONOTTOUCH.FEETYPE(FEECODE)
      ,CREDITCARD VARCHAR2(20)
      ,CUSTID NUMBER
      ,CONSTRAINT INVOICE_CUST_FK FOREIGN KEY (CREDITCARD, CUSTID)
REFERENCES DONOTTOUCH.CUSTCREDIT(CREDITCARDNO, CUSTOMERID)
      ,CONSTRAINT INVOICE_PK PRIMARY KEY (INVOICENO, ITINERARYNUM)
            USING INDEX (CREATE INDEX INVOICE_INDX ON
DONOTTOUCH.INVOICE(INVOICENO, ITINERARYNUM)
            TABLESPACE INDX
            STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10)))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);

CREATE TABLE DONOTTOUCH.TRIPTRAVELLER
      (ITINERARYNUM INT CONSTRAINT TRIPTRAVELLER_ITINERARY_FK
REFERENCES DONOTTOUCH.TRIP(ITINERARYNUM)
      ,TRAVELLERNAME VARCHAR2(50)
      ,CONSTRAINT TRIPTRAVELLER_PK PRIMARY KEY (ITINERARYNUM,
TRAVELLERNAME)
            USING INDEX (CREATE INDEX TRIPTRAVELLER_INDX ON
DONOTTOUCH.TRIPTRAVELLER(ITINERARYNUM, TRAVELLERNAME)
            TABLESPACE INDX
            STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10)))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);


CREATE TABLE DONOTTOUCH.PRODUCT
      (PRODUCTCATNO NUMBER(3) CONSTRAINT PRODUCT_PRODCATNO_FK
REFERENCES DONOTTOUCH.PRODUCTCAT(PRODUCTCATNO)
      ,SUPPLIERNO INT
      ,company varchar2(100)
      ,TAXTYPE NUMBER(3) CONSTRAINT PRODUCT_TAXTYPE_FK REFERENCES
DONOTTOUCH.TAXTYPE(TAXTYPE)
      ,TAXRATE NUMBER(3,2)
      ,PRICE NUMBER
      ,CONSTRAINT product_fk foreign key (supplierno, company)
references DONOTTOUCH.SUPPLIER(supplierno, companyname)
      ,CONSTRAINT PRODUCT_PK PRIMARY KEY (PRODUCTCATNO, SUPPLIERNO,
company)
      USING INDEX (CREATE INDEX PRODUCT_INDX ON
DONOTTOUCH.PRODUCT(PRODUCTCATNO, SUPPLIERNO, company)
      TABLESPACE INDX
      STORAGE (INITIAL 20K NEXT 20K PCTINCREASE 10)))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);


CREATE TABLE DONOTTOUCH.TRIPPRODUCT
      (LISTNO INT DEFAULT donottouch.tripprodseq.nextval CONSTRAINT
TRIPPRODUCT_PK PRIMARY KEY
           USING INDEX (CREATE INDEX TRIPPRODUCT_INDX ON
DONOTTOUCH.TRIPPRODUCT(LISTNO)
           TABLESPACE INDX
           STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10))
      ,ITINERARYNUM INT CONSTRAINT TRIPPROD_ITINERARY_FK REFERENCES
DONOTTOUCH.TRIP(ITINERARYNUM)
      ,SUPPLIERNO INT
      ,company varchar2(100)
      ,PRODUCTCATNO NUMBER(3) CONSTRAINT TRIPPROD_PRODCAT_FK REFERENCES
DONOTTOUCH.PRODUCTCAT(PRODUCTCATNO)
      ,STARTDATE DATE CONSTRAINT TRIPPROD_START_NN NOT NULL
      ,ENDDATE DATE CONSTRAINT TRIPPROD_END_NN NOT NULL
      ,BOOKINGNUM VARCHAR2(15) CONSTRAINT TRIPPROD_BOOKING_NN NOT NULL
      ,DESTINATIONCODE VARCHAR2(5) CONSTRAINT TRIPPROD_DEST_FK
REFERENCES DONOTTOUCH.DESTINATION(DESTINATIONCODE)
      ,CLASSCODE VARCHAR2(5) CONSTRAINT TRIPPROD_CLASS_FK REFERENCES
DONOTTOUCH.CLASS(CLASSCODE)
      ,DESCRIPTIONNO INT CONSTRAINT TRIPPROD_DESC_FK REFERENCES
DONOTTOUCH.DESCRIPTION(DESCRIPTIONNO)
      ,TRIPDESTINATION varchar2(50)
      ,CONSTRAINT TRIPPROD_SUPPLIER_FK FOREIGN KEY (supplierno,company)
REFERENCES DONOTTOUCH.SUPPLIER(SUPPLIERNO,companyname)
      ,CONSTRAINT TRIPPROD_STARTDATE_CK CHECK (STARTDATE <= ENDDATE))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);


CREATE TABLE DONOTTOUCH.COMMISSION
      (LISTNO INT
      ,COMMISSIONNO NUMBER(2) CONSTRAINT COMMISSION_TYPE_FK REFERENCES
DONOTTOUCH.COMMISSIONTYPE(COMMISSIONNO)
      ,COMMISSIONAMOUNT NUMBER(10,2) CONSTRAINT COMMISSION_AMOUNT_NN
NOT NULL
      ,EMPID INT CONSTRAINT COMM_EMP_FK REFERENCES
DONOTTOUCH.EMPLOYEE(EMPID)
      ,CONSTRAINT COMMISSION_PK_FK FOREIGN KEY (LISTNO) REFERENCES
TRIPPRODUCT(LISTNO)
      ,CONSTRAINT COMMISSION_PK PRIMARY KEY (LISTNO, EMPID)
           USING INDEX (CREATE INDEX COMMISSION_INDX ON
DONOTTOUCH.COMMISSION(LISTNO, EMPID)
           TABLESPACE INDX
           STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10)))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);

CREATE TABLE DONOTTOUCH.OPTIONALCUSTDETAIL
```

```
      (CUSTOMERID INT CONSTRAINT CUSTDETAIL_FK REFERENCES
DONOTTOUCH.CUSTOMER(CUSTOMERID)
      ,BIRTHDATE DATE
      ,WEDDING DATE
      ,EMAIL VARCHAR2(50)
      ,WHEELCHAIR CHAR(1) default 'N'
      ,CONSTRAINT OPTCUSTDET_PK PRIMARY KEY (CUSTOMERID)
           USING INDEX (CREATE INDEX OPTCUSTDET_INDX ON
DONOTTOUCH.OPTIONALCUSTDETAIL(CUSTOMERID)
           TABLESPACE INDX
           STORAGE (INITIAL 20K NEXT 10K PCTINCREASE 10)))
      TABLESPACE USERDATA01
      STORAGE (INITIAL 50K NEXT 5K PCTINCREASE 10);
```

Create Users & Profiles Script

```
DROP PROFILE jr_agent CASCADE;
DROP PROFILE int_agent CASCADE;
DROP PROFILE sr_agent CASCADE;
DROP PROFILE manager CASCADE;
DROP PROFILE dba CASCADE;
DROP PROFILE owner CASCADE;




ALTER PROFILE DEFAULT LIMIT
      CONNECT_TIME 480
      IDLE_TIME 5
      FAILED_LOGIN_ATTEMPTS 5
      PASSWORD_LIFE_TIME 90
      PASSWORD_LOCK_TIME 1/24
;
CREATE PROFILE jr_agent LIMIT
      SESSIONS_PER_USER 2
      IDLE_TIME 10
      CPU_PER_SESSION 3000
      CPU_PER_CALL 3000
;
CREATE PROFILE int_agent LIMIT
      SESSIONS_PER_USER 2
      IDLE_TIME 10
      CPU_PER_SESSION 3000
      CPU_PER_CALL 300
;
CREATE PROFILE sr_agent LIMIT
      SESSIONS_PER_USER 2
      CPU_PER_SESSION 3000
      CPU_PER_CALL 6000
      IDLE_TIME 15
;
CREATE PROFILE manager LIMIT
```

```
      SESSIONS_PER_USER 3
      IDLE_TIME 10
      CPU_PER_SESSION 3000
      CPU_PER_CALL 6000
;
CREATE PROFILE dba LIMIT
      SESSIONS_PER_USER 5
      IDLE_TIME UNLIMITED
      CPU_PER_SESSION 3000
      CPU_PER_CALL 6000
      PASSWORD_LOCK_TIME 1/48
;
CREATE PROFILE owner LIMIT
      SESSIONS_PER_USER 2
      IDLE_TIME 15
      CPU_PER_SESSION 3000
      CPU_PER_CALL 3000
      PASSWORD_LIFE_TIME 200
      PASSWORD_LOCK_TIME 1/48
;

--clean up users
DROP USER DONOTTOUCH CASCADE;

DROP USER THEDBA CASCADE;
DROP USER OWNER CASCADE;
DROP USER SR_AGENT CASCADE;
DROP USER INT_AGENT CASCADE;
DROP USER JR_AGENT CASCADE;
DROP USER SUPPORT CASCADE;
DROP USER COM_SPECIAL CASCADE;
DROP USER MANAGER CASCADE;

--create users
CREATE USER DONOTTOUCH IDENTIFIED BY oracle
DEFAULT TABLESPACE userdata01
TEMPORARY TABLESPACE temp01
QUOTA 500m ON userdata01
QUOTA 500m ON userdata02
QUOTA 300m ON indx
QUOTA 300m ON ADHOC
PROFILE dba
ACCOUNT LOCK;

CREATE USER THEDBA IDENTIFIED BY "Oracle"
DEFAULT TABLESPACE userdata01
TEMPORARY TABLESPACE temp01
QUOTA 100m ON ADHOC
PROFILE dba;


CREATE USER OWNER IDENTIFIED BY "Oracle"
```

```
DEFAULT TABLESPACE userdata02
TEMPORARY TABLESPACE temp01
PROFILE owner
PASSWORD EXPIRE;

CREATE USER SR_AGENT IDENTIFIED BY "Oracle"
DEFAULT TABLESPACE userdata02
TEMPORARY TABLESPACE temp01
PROFILE sr_agent
PASSWORD EXPIRE;

CREATE USER INT_AGENT IDENTIFIED BY "Oracle"
DEFAULT TABLESPACE userdata02
TEMPORARY TABLESPACE temp01
PROFILE int_agent
PASSWORD EXPIRE;

CREATE USER JR_AGENT IDENTIFIED BY "Oracle"
DEFAULT TABLESPACE userdata02
TEMPORARY TABLESPACE temp01
PROFILE jr_agent
PASSWORD EXPIRE;

CREATE USER SUPPORT IDENTIFIED BY "Oracle"
DEFAULT TABLESPACE userdata02
TEMPORARY TABLESPACE temp01
PROFILE int_agent
PASSWORD EXPIRE;

CREATE USER MANAGER IDENTIFIED BY "Oracle"
DEFAULT TABLESPACE userdata02
TEMPORARY TABLESPACE temp01
PROFILE manager
PASSWORD EXPIRE;

CREATE USER COM_SPECIAL IDENTIFIED BY "Oracle"
DEFAULT TABLESPACE userdata02
TEMPORARY TABLESPACE temp01
PROFILE manager
PASSWORD EXPIRE;

GRANT CREATE SESSION, CREATE TABLE, CREATE SEQUENCE TO DONOTTOUCH;
```

Create Role Script

```
CONN SYS/Elcaro1! as sysdba;
DROP VIEW DONOTTOUCH.EMP_VIEW CASCADE CONSTRAINTS;

CREATE VIEW DONOTTOUCH.EMP_VIEW AS
     SELECT EMPID, LASTNAME||', '||FIRSTNAME as NAME, PHONE, EMAIL
     FROM DONOTTOUCH.EMPLOYEE
     WITH READ ONLY;
```

```
DROP ROLE TEADBA_R;
DROP ROLE JR_AGENT_R;
DROP ROLE INT_AGENT_R;
DROP ROLE SR_AGENT_R;
DROP ROLE MANAGER_R;
DROP ROLE COM_SPECIAL_R;
DROP ROLE OWNER_R;
DROP ROLE SUPPORT_R;

CREATE ROLE TEADBA_R;
CREATE ROLE JR_AGENT_R;
CREATE ROLE INT_AGENT_R;
CREATE ROLE SR_AGENT_R;
CREATE ROLE MANAGER_R;
CREATE ROLE COM_SPECIAL_R;
CREATE ROLE OWNER_R;
CREATE ROLE SUPPORT_R;

--DBA
GRANT CREATE SESSION, DBA TO TEADBA_R;

GRANT SYSDBA TO THEDBA;
GRANT TEADBA_R TO THEDBA;

--OWNER
GRANT CREATE SESSION TO OWNER_R;
GRANT SELECT ON DONOTTOUCH.AFFILIATION TO OWNER;
GRANT SELECT ON DONOTTOUCH.CLASS TO OWNER;
GRANT SELECT ON DONOTTOUCH.COMMISSION TO OWNER;
GRANT SELECT ON DONOTTOUCH.COMMISSIONTYPE TO OWNER;
GRANT SELECT ON DONOTTOUCH.CREDITCARD TO OWNER;
GRANT SELECT ON DONOTTOUCH.CUSTOMER TO OWNER;
GRANT SELECT ON DONOTTOUCH.CUSTREWARDCARD TO OWNER;
GRANT SELECT ON DONOTTOUCH.CUSTCREDIT TO OWNER;
GRANT SELECT ON DONOTTOUCH.DESTINATION TO OWNER;
GRANT SELECT ON DONOTTOUCH.DESCRIPTION TO OWNER;
GRANT SELECT ON DONOTTOUCH.EMPLOYEE TO OWNER;
GRANT SELECT ON DONOTTOUCH.FEETYPE TO OWNER;
GRANT SELECT ON DONOTTOUCH.INVOICE TO OWNER;
GRANT SELECT ON DONOTTOUCH.LOCATION TO OWNER;
GRANT SELECT ON DONOTTOUCH.OPTIONALCUSTDETAIL TO OWNER;
GRANT SELECT ON DONOTTOUCH.PAYMENTTYPE TO OWNER;
GRANT SELECT ON DONOTTOUCH.PRODUCT TO OWNER;
GRANT SELECT ON DONOTTOUCH.PRODUCTCAT TO OWNER;
GRANT SELECT ON DONOTTOUCH.REWARDCARD TO OWNER;
GRANT SELECT ON DONOTTOUCH.SUPPLIER TO OWNER;
GRANT SELECT ON DONOTTOUCH.TAXTYPE TO OWNER;
GRANT SELECT ON DONOTTOUCH.TRIPPRODUCT TO OWNER;
GRANT SELECT ON DONOTTOUCH.TRIPTYPE TO OWNER;
GRANT SELECT ON DONOTTOUCH.TRIP TO OWNER;
GRANT SELECT ON DONOTTOUCH.TRIPTRAVELLER TO OWNER;
```

```
--SUPPORT STAFF
GRANT CREATE SESSION TO SUPPORT_R;

GRANT SELECT ON DONOTTOUCH.CUSTOMER TO SUPPORT_R;
GRANT SELECT ON DONOTTOUCH.OPTIONALCUSTDETAIL TO SUPPORT_R;
GRANT SELECT ON DONOTTOUCH.CUSTREWARDCARD TO SUPPORT_R;
GRANT SELECT ON DONOTTOUCH.EMP_VIEW TO SUPPORT_R;

--JR AGENT
GRANT SUPPORT_R TO JR_AGENT_R;

GRANT SELECT ON DONOTTOUCH.SUPPLIER TO JR_AGENT_R;
GRANT SELECT ON DONOTTOUCH.PRODUCT TO JR_AGENT_R;
GRANT SELECT ON DONOTTOUCH.TAXTYPE TO JR_AGENT_R;
GRANT SELECT ON DONOTTOUCH.PRODUCTCAT TO JR_AGENT_R;
GRANT SELECT, INSERT ON DONOTTOUCH.TRIPPRODUCT TO JR_AGENT_R;
GRANT SELECT ON DONOTTOUCH.DESCRIPTION TO JR_AGENT_R;
GRANT SELECT ON DONOTTOUCH.LOCATION TO JR_AGENT_R;
GRANT SELECT, INSERT ON DONOTTOUCH.TRIP TO JR_AGENT_R;
GRANT SELECT, INSERT, UPDATE, DELETE ON DONOTTOUCH.TRIPTRAVELLER TO
JR_AGENT_R;
GRANT SELECT ON DONOTTOUCH.FEETYPE TO JR_AGENT_R;
GRANT SELECT, INSERT ON DONOTTOUCH.CUSTREWARDCARD TO JR_AGENT_R;
GRANT SELECT ON DONOTTOUCH.REWARDCARD TO JR_AGENT_R;
GRANT SELECT ON DONOTTOUCH.DESTINATION TO JR_AGENT_R;
GRANT SELECT ON DONOTTOUCH.CLASS TO JR_AGENT_R;

GRANT INSERT ON DONOTTOUCH.COMMISSION TO JR_AGENT_R;
GRANT SELECT ON DONOTTOUCH.COMMISSIONTYPE TO JR_AGENT_R;

GRANT INSERT, UPDATE ON DONOTTOUCH.CUSTOMER TO JR_AGENT_R;
GRANT INSERT, UPDATE ON DONOTTOUCH.OPTIONALCUSTDETAIL TO JR_AGENT_R;

GRANT INSERT ON DONOTTOUCH.CUSTCREDIT TO JR_AGENT_R;

--INT AGENT, assuming agents that become intermediates will be the
ones to start generating invoices.
GRANT JR_AGENT_R TO INT_AGENT_R;
GRANT UPDATE ON DONOTTOUCH.CUSTREWARDCARD TO INT_AGENT_R;
GRANT INSERT ON DONOTTOUCH.REWARDCARD TO INT_AGENT_R;
GRANT SELECT ON DONOTTOUCH.CREDITCARD TO INT_AGENT_R;
GRANT SELECT ON DONOTTOUCH.CUSTCREDIT TO INT_AGENT_R;

GRANT SELECT, INSERT ON DONOTTOUCH.INVOICE TO INT_AGENT_R;
GRANT SELECT ON DONOTTOUCH.PAYMENTTYPE TO INT_AGENT_R;

GRANT UPDATE ON DONOTTOUCH.TRIPPRODUCT TO INT_AGENT_R;
GRANT UPDATE ON DONOTTOUCH.TRIP TO INT_AGENT_R;
GRANT INSERT ON DONOTTOUCH.DESCRIPTION TO INT_AGENT_R;
```

```
--SR AGENT
GRANT INT_AGENT_R TO SR_AGENT_R;

GRANT UPDATE ON DONOTTOUCH.INVOICE TO SR_AGENT_R;
GRANT UPDATE ON DONOTTOUCH.CUSTCREDIT TO SR_AGENT_R;
GRANT UPDATE ON DONOTTOUCH.SUPPLIER TO SR_AGENT_R;
GRANT UPDATE ON DONOTTOUCH.PRODUCT TO SR_AGENT_R;


--COMMISSION SPECIALIST
GRANT SR_AGENT_R TO COM_SPECIAL_R;

GRANT SELECT, UPDATE, INSERT, DELETE ON DONOTTOUCH.COMMISSION TO
COM_SPECIAL_R;

--MANAGER
GRANT SR_AGENT_R TO MANAGER_R;
GRANT COM_SPECIAL_R TO MANAGER_R;

GRANT SELECT, INSERT, UPDATE, DELETE ON DONOTTOUCH.EMPLOYEE TO
MANAGER_R;
GRANT UPDATE, INSERT, DELETE  ON DONOTTOUCH.SUPPLIER TO MANAGER_R;
GRANT UPDATE, INSERT, DELETE  ON DONOTTOUCH.TAXTYPE TO MANAGER_R;
GRANT UPDATE, INSERT, DELETE  ON DONOTTOUCH.PRODUCT TO MANAGER_R;
GRANT UPDATE, INSERT, DELETE  ON DONOTTOUCH.DESTINATION TO MANAGER_R;
GRANT UPDATE, INSERT, DELETE  ON DONOTTOUCH.COMMISSION TO MANAGER_R;
GRANT UPDATE, INSERT, DELETE  ON DONOTTOUCH.LOCATION TO MANAGER_R;
GRANT UPDATE, INSERT, DELETE  ON DONOTTOUCH.PRODUCT TO MANAGER_R;


--Granting everyone their roles....
GRANT OWNER_R TO  OWNER;
GRANT SR_AGENT_R TO SR_AGENT;
GRANT INT_AGENT_R TO INT_AGENT;
GRANT JR_AGENT_R TO JR_AGENT;
GRANT SUPPORT_R TO SUPPORT;
GRANT MANAGER_R TO MANAGER;
GRANT COM_SPECIAL_R TO COM_SPECIAL;


COL USERNAME FOR A20
COL STATUS FOR A20
COL 'DEFAULT TS' FOR A15
COL 'TMP TS' FOR A8
COL 'LAST LOG' FOR A12
col profile for a15
SET LINESIZE 100
SET PAGESIZE 50
TTITLE 'Users'
SELECT USERNAME, ACCOUNT_STATUS STATUS,
 DEFAULT_TABLESPACE "DEFAULT TS", TEMPORARY_TABLESPACE "TMP TS",
to_char(LAST_LOGIN,'MON DD, YYYY') "LAST LOG", profile
```

```
FROM DBA_USERS
ORDER BY 2 , 1;


TTITLE 'PRIVILEGES'
col grantee for a15
col owner for a10
col table_name for a15 heading "Table" word_wrapped
col privilege for a15
select grantee, owner, table_name, privilege
FROM dba_tab_privs
where OWNER = 'DONOTTOUCH'
order by 1,3;


TTITLE 'Granted Roles'
COL GRANTED_ROLE FOR A20
select grantee, granted_role
from dba_role_privs
where grantee IN ('JR_AGENT', 'SR_AGENT','OWNER','INT_AGENT',
'SUPPORT', 'MANAGER','COM_SPECIAL');


ttitle 'Profiles'
col profile for a20 word_wrapped
col resource_name for a32 heading "Resource" word_wrapped
col limit for a10 word_wrapped
col resource_type for a8
break on profile
select profile, resource_name, resource_type, limit
FROM dba_profiles
order by 1;
```

Audit Script

```
--Set the audit_trail = dba, extended in pfile

AUDIT SELECT, DELETE ON SYS.AUD$ BY ACCESS;

--Audit failures of select and deletes of tables for suspicious
activity
AUDIT SELECT TABLE, DELETE TABLE, EXECUTE PROCEDURE BY ACCESS WHENEVER
NOT SUCCESSFUL;


AUDIT SELECT, UPDATE, DELETE ON DONOTTOUCH.CUSTCREDIT BY ACCESS;

AUDIT INSERT, UPDATE, DELETE ON DONOTTOUCH.INVOICE BY ACCESS;

AUDIT UPDATE, DELETE ON DONOTTOUCH.PRODUCT;
AUDIT DELETE ON DONOTTOUCH.TRIPPRODUCT;
AUDIT DELETE ON DONOTTOUCH.TRIP;

AUDIT ALTER, GRANT, INSERT, UPDATE, DELETE, SELECT ON
DONOTTOUCH.COMMISSION BY ACCESS;

AUDIT UPDATE, DELETE ON DONOTTOUCH.SUPPLIER BY SESSION;
```

Loader Scripts

Customers

```
OPTIONS(skip=1, direct=y)
LOAD DATA
     INFILE 'c:\tea\projectdata_cust.csv'
     BADFILE 'c:\tea\bad_cust.csv'
     DISCARDFILE 'c:\tea\discard_supp.csv'
     TRUNCATE INTO TABLE DONOTTOUCH.cust_adhoc
     Fields terminated by "," optionally enclosed by '"'
     (
          cust_id integer external
          ,first_name char
          ,last_name char
          ,agent char
          ,email char
          ,home_phone char
          ,business_phone char
          ,birthdate date 'YYYY-MM-DD'
          ,address char
          ,city char
          ,postal char
          ,province char
          ,country char
     )
```

Sales

```
OPTIONS(skip=1, direct=y)
LOAD DATA
     INFILE 'c:\tea\projectdata_sale.csv'
     BADFILE 'c:\tea\bad_sale.csv'
     DISCARDFILE 'c:\tea\discard.csv'
     TRUNCATE INTO TABLE DONOTTOUCH.sale_adhoc
     Fields terminated by "," optionally enclosed '"'
     (
     sales_date date 'YYYY-MM-DD'
     ,cust_id integer external
     ,itinerary char
     ,agent char
     ,booking char
     ,prod_cat integer external
     ,supplier_id integer external
     ,supp_off integer external
     ,trip_start date 'YYYY-MM-DD'
     ,trip_end date 'YYYY-MM-DD'
     ,class char
     ,num_travellers integer external
     ,product char
     ,description char
     ,destination char
     ,dest_id char
     ,credit_card char
     ,expiry date 'YYYY-MM-DD'
     ,card_num char
     ,bill_date date 'YYYY-MM-DD'
     ,bill_desc char
     ,base_price float external
     ,tot_price float external
     ,bill_amt float external
     ,agency_fee_code char
     ,agency_fee_amt float external
     ,agency_comm float external
     )
```

Suppliers

```
OPTIONS(skip=1, direct=y)
LOAD DATA
     INFILE 'c:\tea\projectdata_supp.csv'
     BADFILE 'c:\tea\bad_supp.csv'
     DISCARDFILE 'c:\tea\discard.csv'
     TRUNCATE INTO TABLE DONOTTOUCH.supp_adhoc
     Fields terminated by "," optionally enclosed '"'
     (
          prod_supp integer external
          ,prod_cat integer external
          ,supp_off integer external
          ,prod_desc char
          ,cont_name char
          ,company char
          ,address char
          ,address2 char
          ,city char
          ,prov char
          ,postal char
          ,country char
          ,phone char
          ,fax char
          ,email char
          ,web char
          ,represents char
          ,affiliation char
     )
```

Data Migrating Scripts

```
--Manipulating data
--Customers
```

```
--inserting all but birthdate, agent, city
--Resetting the tables...
DELETE FROM DONOTTOUCH.OPTIONALCUSTDETAIL CASCADE;
DELETE FROM DONOTTOUCH.CUSTOMER CASCADE;
--Customer
INSERT INTO DONOTTOUCH.CUSTOMER
(CUSTOMERID, FIRSTNAME, LASTNAME, HPHONE, BPHONE, ADDRESS, ZIP_POSTAL)
(SELECT CUST_ID, FIRST_NAME, LAST_NAME, HOME_PHONE, BUSINESS_PHONE,
ADDRESS, POSTAL
           FROM DONOTTOUCH.CUST_ADHOC);
--Optional Customer
INSERT INTO DONOTTOUCH.OPTIONALCUSTDETAIL
(CUSTOMERID, BIRTHDATE, EMAIL)
(SELECT CUST_ID, BIRTHDATE, EMAIL
           FROM DONOTTOUCH.CUST_ADHOC
           WHERE BIRTHDATE IS NOT NULL
           OR
                EMAIL IS NOT NULL);

DECLARE
     CURSOR agentinitial_curs IS
           SELECT A.CUST_ID, E.EMPID
           FROM DONOTTOUCH.CUST_ADHOC A
           JOIN
           (SELECT substr(firstname, 1,1)||substr(lastname, 1,1) as
INITIALS, empid
           FROM DONOTTOUCH.EMPLOYEE) E
           ON A.AGENT = E.INITIALS;
     v_cid number(3);
     v_eid number(3);
BEGIN
     OPEN agentinitial_curs;
     LOOP
           FETCH agentinitial_curs INTO v_cid, v_eid;
           EXIT WHEN agentinitial_curs%NOTFOUND;

           UPDATE DONOTTOUCH.CUSTOMER
           SET EMPID = v_eid
           WHERE CUSTOMERID = v_cid;
     END LOOP;

     CLOSE agentinitial_curs;
END;
/
/*
col CUSTOMERID for 999 HEADING 'ID'
col EMAIL for a30 WORD_WRAPPED
col EMPID for 999
col 'Name' for a30 WORD_WRAPPED
col hphone HEADING "Home Phone"
col bphone HEADING "Business Phone"
col address for a30 WORD_WRAPPED
col zip_postal HEADING "Postal"

TTITLE 'Customers'
SELECT CUSTOMERID, empid, LASTNAME||', '||FIRSTNAME "Name", hphone,
bphone, address, zip_postal
FROM DONOTTOUCH.CUSTOMER
ORDER BY 1
FETCH FIRST 5 ROWS ONLY;

TTITLE 'Customers - Optional Details'
SELECT *
FROM DONOTTOUCH.OPTIONALCUSTDETAIL
ORDER BY 1
FETCH FIRST 5 ROWS ONLY;

SELECT COUNT(EMPID)
FROM DONOTTOUCH.CUSTOMER; */

COMMIT;

--Updates to insert into the locations table
UPDATE DONOTTOUCH.supp_adhoc
SET country = 'Canada'
WHERE prov IN ('AB', 'BC', 'ON', 'SK', 'PQ', 'YT', 'NS', 'MB', 'NF',
'NT', 'PE', 'NB');

UPDATE DONOTTOUCH.supp_adhoc
SET country = 'USA'
WHERE prov IN ('FL', 'CA', 'AZ') AND
      city IN ('Los Angeles', 'Phoenix', 'Miami');
/*
SELECT count(*)
FROM DONOTTOUCH.supp_adhoc
WHERE country is not null
AND city is not null
AND prov is not null; */

--Location
INSERT INTO DONOTTOUCH.LOCATION
(CITYNAME, STATEPROV, COUNTRYNAME)
SELECT DISTINCT UPPER(city), prov, country
FROM DONOTTOUCH.supp_adhoc
WHERE city is not null
ORDER BY 3,2,1;

--As all customers in current data set are from calgary, set all
citycodes for customer as calgary...
UPDATE DONOTTOUCH.CUSTOMER
SET CITYCODE = (SELECT CITYCODE
                      FROM DONOTTOUCH.LOCATION
                      WHERE lower(CITYNAME) = 'calgary');
```

```
COMMIT;

--Filling in Supplier & SupplierDetails
INSERT INTO DONOTTOUCH.SUPPLIER (SUPPLIERNO, COMPANYNAME)
SELECT DISTINCT PROD_SUPP, company
FROM DONOTTOUCH.SUPP_ADHOC
WHERE COMPANY is NOT NULL;


DECLARE
      CURSOR thecursor IS
            SELECT DISTINCT PROD_SUPP, company, SUPP_OFF,
address||address2, cont_name, phone, fax, city, postal
            FROM DONOTTOUCH.SUPP_ADHOC
            WHERE cont_name is not null OR address||address2 is not
null;
      v1 number(38);
      v3 number(3);
      v2 varchar2(100);
      v4 varchar2(100);
      v5 varchar2(50);
      v6 varchar2(15);
      v7 varchar2(15);
      v8 varchar2(50);
      v9 varchar2(10);
      cursor representcur IS
            SELECT represents, prod_supp, company
            from donottouch.supp_adhoc
            where represents is not null;
      v_r1 varchar2(100);
      v_r2 number(38);
      v_r3 varchar2(100);
      cursor affilcur IS
            SELECT affiliation, prod_supp, company
            from donottouch.supp_adhoc
            where affiliation is not null;
      v_a1 varchar(10);
      cursor emailwebcur IS
            SELECT email, web, prod_supp, company
            from donottouch.supp_adhoc
            where email is not null OR web is not null;
      v_w1 varchar(50);
      v_w2 varchar(50);
BEGIN
      OPEN thecursor;
      LOOP
            FETCH thecursor into v1,v2,v3,v4, v5, v6, v7, v8, v9;
            EXIT WHEN thecursor%notfound;

            UPDATE DONOTTOUCH.SUPPLIER
            SET officeno = v3,
                  address = v4,
                  contactname = v5,
                  PHONE = v6,
                  FAX = v7,
                  zip_postal = v9,
                  citycode = (SELECT citycode FROM DONOTTOUCH.LOCATION
WHERE lower(cityname) = lower(v8))
            WHERE SUPPLIERNO = v1 AND
                  companyname = v2;
      END LOOP;
      CLOSE thecursor;

      OPEN representcur;
      LOOP
            FETCH representcur INTO v_r1, v_r2, v_r3;
            EXIT WHEN representcur%notfound;

            UPDATE DONOTTOUCH.SUPPLIER
            SET REPRESENTS = v_r1,
                  REP_NO = v_r2
            WHERE SUPPLIERNO = v_r2 AND
                  COMPANYNAME = v_r3;
      END LOOP;
      CLOSE representcur;

      OPEN affilcur;
      LOOP
            FETCH affilcur INTO v_a1, v_r2, v_r3;
            EXIT WHEN affilcur%notfound;

            UPDATE DONOTTOUCH.SUPPLIER
            SET AFFILIATIONCODE = v_a1
            WHERE SUPPLIERNO = v_r2 AND
                  COMPANYNAME = v_r3;
      END LOOP;
      CLOSE affilcur;

      OPEN emailwebcur;
      LOOP
            FETCH emailwebcur INTO v_w1,v_w2, v_r2, v_r3;
            EXIT WHEN emailwebcur%notfound;

            UPDATE DONOTTOUCH.SUPPLIER
            SET WEB = v_w2,
                  EMAIL = v_w1
            WHERE SUPPLIERNO = v_r2 AND
                  COMPANYNAME = v_r3;
      END LOOP;
      CLOSE emailwebcur;
END;
/
commit;
```

```
--Fill out description  table
INSERT INTO DONOTTOUCH.DESCRIPTION (DESCRIPTION)
SELECT DISTINCT upper(description)
FROM DONOTTOUCH.sale_adhoc
WHERE description is not null
ORDER BY 1;

commit;

--Inserting into Creditcard

INSERT INTO DONOTTOUCH.CREDITCARD (CREDITDESC)
SELECT DISTINCT UPPER(credit_card)
FROM donottouch.sale_adhoc
WHERE credit_card IS NOT NULL;

col CARDTYPENO for 999 HEADING 'No.'
col CREDITDESC for a30 HEADING 'Description' WORD_WRAPPED

TTITLE 'Credit Card Types'
SELECT *
FROM DONOTTOUCH.CREDITCARD
ORDER BY 1;

COMMIT;

INSERT INTO DONOTTOUCH.CUSTCREDIT (customerid, creditcardno)
SELECT distinct cust_id, card_num
FROM DONOTTOUCH.sale_adhoc;

DECLARE
      cursor creditcurs is
            SELECT distinct cust_id, card_num, credit_card, expiry
            FROM donottouch.sale_adhoc
            order by expiry asc;
      v1 number;
      v2 varchar2(20);
      v3 varchar2(20);
      v4 date;
BEGIN
      OPEN creditcurs;
      LOOP
            FETCH creditcurs into v1, v2, v3, v4;
            EXIT WHEN creditcurs%notfound;

            UPDATE DONOTTOUCH.CUSTCREDIT
            SET EXPDATE = v4,
                  CARDTYPENO = (select cardtypeno from
donottouch.creditcard where creditdesc = v3)
            WHERE customerid = v1 and CREDITCARDNO = v2;

      END LOOP;
      CLOSE creditcurs;
END;
/


--Inserting into CustRewardCard
DELETE FROM DONOTTOUCH.CUSTREWARDCARD

INSERT INTO DONOTTOUCH.CUSTREWARDCARD
VALUES ('110202', 1, 347);

INSERT INTO DONOTTOUCH.CUSTREWARDCARD
VALUES ('112202', 2, 140);

COMMIT;
--Insert products that have prices - the most recent prices
INSERT into DONOTTOUCH.PRODUCT (PRODUCTCATNO, SUPPLIERNO, COMPANY)
SELECT distinct s.prod_cat, s.supplier_id, p.company
FROM donottouch.sale_adhoc s join donottouch.supp_adhoc p
on p.prod_supp = s.supplier_id and p.prod_cat = s.prod_cat;

DECLARE
      CURSOR pricecur IS
            SELECT DISTINCT s.prod_cat, s.supplier_id, p.company,
s.base_price, s.sales_date
            FROM donottouch.sale_adhoc s join donottouch.supp_adhoc p
            on p.prod_supp = s.supplier_id and p.prod_cat = s.prod_cat
            ORDER BY s.SALES_DATE ASC;
      v1 number;
      v2 number;
      v3 varchar2(100);
      v4 number;
      v5 date;
BEGIN
      open pricecur;
      LOOP
            FETCH pricecur INTO v1, v2, v3, v4,v5;
            EXIT WHEN pricecur%notfound;

            UPDATE DONOTTOUCH.product
            SET price = v4
            WHERE productcatno = v1 AND
            SUPPLIERNO = v2 AND
            COMPANY = v3;

      END LOOP;
      close pricecur;
END;
/

--Inserts into TRIP
INSERT INTO DONOTTOUCH.TRIP
```

```
(itinerarynum)
SELECT DISTINCT ITINERARY
FROM DONOTTOUCH.sale_adhoc
order by 1;

DECLARE
     CURSOR tripcur IS
          SELECT distinct itinerary, cust_id, num_travellers,
trip_start, trip_end
          FROM DONOTTOUCH.sale_adhoc
          ORDER BY itinerary;
     v1 varchar2(20);
     v2 number;
     v3 number;
     v4 date;
     v5 date;
BEGIN
     open tripcur;
     LOOP
          FETCH tripcur INTO v1, v2, v3,v4, v5;
          EXIT WHEN tripcur%notfound;

          UPDATE DONOTTOUCH.TRIP
          SET
               customerid = v2,
               startdate = v4,
               enddate = v5,
               travellers = v3
          WHERE itinerarynum = v1 and customerid is null;

          UPDATE DONOTTOUCH.TRIP
          SET
               customerid = v2,
               startdate = v4
          WHERE startdate > v4 and itinerarynum = v1;

          UPDATE DONOTTOUCH.TRIP
          SET
               customerid = v2,
               enddate = v5
          WHERE enddate < v5 and itinerarynum = v1;
     END LOOP;
     close tripcur;
END;
/


--Inserts into TripProduct
INSERT INTO DONOTTOUCH.TRIPPRODUCT (ITINERARYNUM, BOOKINGNUM,
STARTDATE, ENDDATE,
     DESTINATIONCODE, TRIPDESTINATION, classcode)
```

```
SELECT distinct itinerary, booking, trip_start, trip_end, DEST_ID,
destination,class
FROM DONOTTOUCH.SALE_ADHOC
ORDER BY 1;

DECLARE
     CURSOR bookingcurs IS
          SELECT distinct BOOKING, prod_cat, supplier_id
          FROM DONOTTOUCH.sale_adhoc;
     book varchar2(15);
     prod number;
     supp number;
     des varchar2(100);
     cursor bookingcurs2 is select distinct booking, description
     from donottouch.sale_adhoc;

BEGIN
     open bookingcurs;
     LOOP
          FETCH bookingcurs INTO book, prod, supp;
          EXIT WHEN bookingcurs%notfound;

          UPDATE DONOTTOUCH.TRIPPRODUCT
          SET
               supplierno = supp,
               productcatno = prod,
               company = (SELECT company
                              FROM DONOTTOUCH.product
                              WHERE productcatno = prod and
                              supplierno = supp)
          where BOOKINGNUM = book;
     END LOOP;
     close bookingcurs;
     open bookingcurs2;
     LOOP
          FETCH bookingcurs2 INTO book, des;
          EXIT WHEN bookingcurs2%notfound;

          UPDATE DONOTTOUCH.TRIPPRODUCT
          SET descriptionno = (SELECT descriptionno
                              FROM DONOTTOUCH.description
                              WHERE lower(description) = lower(des))
          WHERE bookingnum = book;
     END LOOP;
     close bookingcurs2;
END;
/

--Inserts into Commission

DECLARE
     cursor commcurs IS
```

```
            SELECT distinct itinerary, booking, agent, agency_comm
            from donottouch.sale_adhoc
            where agent is not null
            order by 1;

      v1 varchar2(20);
      v2 varchar2(20);
      v3 varchar2(3);
      v4 number;
      lno number;
      empno number;
BEGIN
      open commcurs;
      LOOP
            FETCH commcurs into v1, v2, v3, v4;
            EXIT WHEN commcurs%notfound;
            select listno into lno from donottouch.tripproduct where
bookingnum = v2 and itinerarynum = v1 fetch first 1 row only;

            if upper(v3) = 'BS' then
                  empno := 7;
            elsif upper(v3) = 'JW' then
                  empno := 5;
            else
            select empid into empno from donottouch.employee where (
                                       lower(substr(firstname,
1,1)||substr(lastname, 1,1))=lower(v3))
                                       fetch first 1 row only;
            end if;
            INSERT INTO DONOTTOUCH.COMMISSION
            values(lno, 6, v4, empno);

      END LOOP;
      close commcurs;
exception when DUP_VAL_ON_INDEX THEN
      UPDATE donottouch.commission
      SET COMMISSIONAMOUNT=v4
      where listno = lno and empid=empno;
END;
/
commit;

--Inserts into Invoice

DECLARE
      CURSOR invcurs is
            SELECT itinerary, bill_desc, agency_fee_code, sales_date,
bill_date,
                  bill_amt, (tot_price+agency_fee_amt),cust_id, card_num
            FROM donottouch.sale_adhoc
            order by 1;
      v1 number;
      v2 varchar2(50);
      v3 varchar2(10);
      v4 date;
      v5 date;
      v6 number;
      v7 number;
      v9 number;
      v8 varchar2(20);
BEGIN
      open invcurs;

      LOOP
           FETCH invcurs into v1, v2, v3, v4, v5, v6, v7,v9,v8;
           EXIT WHEN invcurs%notfound;

           INSERT INTO donottouch.invoice
           (itinerarynum, paymenttype, billingdate, billamount,
remainder, saledate, feecode, creditcard, custid)
           values (v1, (select paymenttype from donottouch.paymenttype
where paymentdesc = v2), v5, v6, v7-v6,v4,v3,v8,v9);
      END LOOP;
      close invcurs;
END;
/

--insert into trip traveller
INSERT INTO DONOTTOUCH.TRIPTRAVELLER
values (584,'John Doe');

commit;
```

Additional Data Insert Script

```
/*
This script inserts the data for 11/25 tables
*/
set echo off;
--Resetting sequences...
DROP SEQUENCE donottouch.empidcount;
DROP SEQUENCE donottouch.paymenttypeseq;
DROP SEQUENCE donottouch.taxtypeseq;
DROP SEQUENCE donottouch.rewardcardseq;


CREATE SEQUENCE donottouch.rewardcardseq
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;

CREATE SEQUENCE donottouch.empidcount
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;
CREATE SEQUENCE donottouch.taxtypeseq
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;
CREATE SEQUENCE donottouch.paymenttypeseq
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;

--Inserting Values into Productcat Table
DELETE FROM DONOTTOUCH.PRODUCTCAT CASCADE;

DECLARE
      counter number:=100;
BEGIN

      LOOP
            IF counter < 150 THEN
                  INSERT INTO DONOTTOUCH.PRODUCTCAT
                  VALUES (counter, 'Airline Consolidators');
            ELSIF counter < 200 THEN
                  INSERT INTO DONOTTOUCH.PRODUCTCAT
                  VALUES (counter, 'Airlines');
            ELSIF counter < 250 THEN
                  INSERT INTO DONOTTOUCH.PRODUCTCAT
                  VALUES (counter, 'Attractions');
            ELSIF counter < 300 THEN
                  INSERT INTO DONOTTOUCH.PRODUCTCAT
                  VALUES (counter, 'Car Rentals');
            ELSIF counter < 350 THEN
                  INSERT INTO DONOTTOUCH.PRODUCTCAT
                  VALUES (counter, 'Cruise Lines');
            ELSIF counter < 400 THEN
                  INSERT INTO DONOTTOUCH.PRODUCTCAT
                  VALUES (counter, 'Hotel Reps and Chains in Canada');
            ELSIF counter < 450 THEN
                  INSERT INTO DONOTTOUCH.PRODUCTCAT
                  VALUES (counter, 'Motor Coach Tour Operators');
            ELSIF counter < 500 THEN
                  INSERT INTO DONOTTOUCH.PRODUCTCAT
                  VALUES (counter, 'Railroads');
            ELSIF counter < 550 THEN
                  INSERT INTO DONOTTOUCH.PRODUCTCAT
                  VALUES (counter, 'Tour Operators/Wholesales');
            ELSIF counter < 600 THEN
                  INSERT INTO DONOTTOUCH.PRODUCTCAT
                  VALUES (counter, 'Travel Insurance');
```

```
           ELSIF counter < 650 THEN
                INSERT INTO DONOTTOUCH.PRODUCTCAT
                VALUES (counter, 'Yacht and Boat Charters');
           ELSE
                EXIT;
           END IF;
           counter := counter + 1;
     END LOOP;

END;
/

/* col PRODUCTCATEGORY for a35 HEADING CATEGORY WORD_WRAPPED
col 'Category Range' for a10 HEADING "Category|Range"
set linesize 50
set pagesize 50

SELECT COUNT(*)
FROM DONOTTOUCH.PRODUCTCAT;

TTITLE 'Product Categories'
SELECT PRODUCTCATNO || ' - '|| lastno "Category Range",
PRODUCTCATEGORY
FROM ( SELECT PRODUCTCATEGORY,
                PRODUCTCATNO,
                min(PRODUCTCATNO) OVER (PARTITION BY PRODUCTCATEGORY)
firstno,
                max(PRODUCTCATNO) OVER (PARTITION BY PRODUCTCATEGORY)
lastno
           FROM DONOTTOUCH.PRODUCTCAT)
WHERE PRODUCTCATNO = firstno
ORDER BY 1; */

COMMIT;

--Inserting into Affiliation
DELETE FROM DONOTTOUCH.AFFILIATION CASCADE;

INSERT INTO DONOTTOUCH.AFFILIATION
VALUES ('ACTA', 'Association of Canadian Travel Agents');

INSERT INTO DONOTTOUCH.AFFILIATION
VALUES ('ACTANEW', '');

INSERT INTO DONOTTOUCH.AFFILIATION
VALUES ('ACTANEWP', '');

INSERT INTO DONOTTOUCH.AFFILIATION
VALUES ('ACTAPGY', '');

INSERT INTO DONOTTOUCH.AFFILIATION
VALUES ('NEW', '');
```

```
INSERT INTO DONOTTOUCH.AFFILIATION
VALUES ('NEWPGY', '');

INSERT INTO DONOTTOUCH.AFFILIATION
VALUES ('PGY', '');

/* col affiliationcode for a8 HEADING 'Code'
col affiliationdesc for a20 HEADING 'Description' WORD_WRAPPED

TTITLE 'Affiliation'
SELECT *
FROM DONOTTOUCH.AFFILIATION
ORDER BY 1; */

COMMIT;
--insert an employee for those empty sales...
INSERT INTO DONOTTOUCH.EMPLOYEE (empid, firstname, lastname)
VALUES (0, 'Bob', 'Manager');
--Inserting into FeeType
DELETE FROM DONOTTOUCH.FEETYPE CASCADE;

INSERT INTO DONOTTOUCH.FEETYPE
VALUES ('RF', 'Refund', 25);

INSERT INTO DONOTTOUCH.FEETYPE
VALUES ('BK', 'Booking Charge', 25);

INSERT INTO DONOTTOUCH.FEETYPE
VALUES ('CH', 'Change', 15);

INSERT INTO DONOTTOUCH.FEETYPE
VALUES ('RS', 'Research', 50);

INSERT INTO DONOTTOUCH.FEETYPE
VALUES ('GR', 'Group Booking', 100);

INSERT INTO DONOTTOUCH.FEETYPE
VALUES ('NSF', 'Insufficient Funds', 25);

INSERT INTO DONOTTOUCH.FEETYPE
VALUES ('NC', 'No Charge', 0);

/* col feecode for a4 HEADING 'Code'
col feetypedesc for a20 HEADING 'Description'
col feeamount for $999 HEADING 'Amount'

TTITLE 'Fees'
SELECT *
FROM DONOTTOUCH.FEETYPE
ORDER BY 3; */
```

```
COMMIT;

--Inserting into CommissionType
DELETE FROM DONOTTOUCH.COMMISSIONTYPE CASCADE;

INSERT INTO DONOTTOUCH.COMMISSIONTYPE
VALUES (1, 'Booking');

INSERT INTO DONOTTOUCH.COMMISSIONTYPE
VALUES (2, 'Paying');

INSERT INTO DONOTTOUCH.COMMISSIONTYPE
VALUES (3, 'Overdue');

INSERT INTO DONOTTOUCH.COMMISSIONTYPE
VALUES (4, 'Invalid');

INSERT INTO DONOTTOUCH.COMMISSIONTYPE
VALUES (5, 'Cancelled');

INSERT INTO DONOTTOUCH.COMMISSIONTYPE
VALUES (6, 'Unknown');
/*
col commissionno for 999 HEADING 'No.'
col commissiontype for a10 HEADING 'Type'

TTITLE 'Commission Types'
SELECT *
FROM DONOTTOUCH.COMMISSIONTYPE
ORDER BY 1; */

COMMIT;

--Inserting Employee
DELETE FROM DONOTTOUCH.EMPLOYEE CASCADE;

INSERT INTO DONOTTOUCH.EMPLOYEE
(FIRSTNAME, LASTNAME)
VALUES ('Janet', 'Delton');

INSERT INTO DONOTTOUCH.EMPLOYEE
(FIRSTNAME, LASTNAME)
VALUES ('Judy', 'Lisle');

INSERT INTO DONOTTOUCH.EMPLOYEE
(FIRSTNAME, LASTNAME)
VALUES ('Dennis C.', 'Reynolds');

INSERT INTO DONOTTOUCH.EMPLOYEE
(FIRSTNAME, LASTNAME)
VALUES ('John', 'Coville');
```

```
INSERT INTO DONOTTOUCH.EMPLOYEE
(FIRSTNAME, LASTNAME)
VALUES ('Janice W.', 'Dahl');

INSERT INTO DONOTTOUCH.EMPLOYEE
(FIRSTNAME, LASTNAME)
VALUES ('Bruce J.', 'Dixon');

INSERT INTO DONOTTOUCH.EMPLOYEE
(FIRSTNAME, LASTNAME)
VALUES ('Beverly S.','Jones');

INSERT INTO DONOTTOUCH.EMPLOYEE
(FIRSTNAME, LASTNAME)
VALUES ('Jane', 'Merrill');

INSERT INTO DONOTTOUCH.EMPLOYEE
(FIRSTNAME, LASTNAME)
VALUES ('Brian S.', 'Peterson');
/*
TTITLE 'Employees'
col ID for 999
col 'Name' for a30 WORD_WRAPPED

SELECT empid ID, lastname||', '||firstname "Name"
FROM DONOTTOUCH.EMPLOYEE
ORDER BY 1; */

COMMIT;

--Inserting into DESTINATION
DELETE FROM DONOTTOUCH.DESTINATION CASCADE;

INSERT INTO DONOTTOUCH.DESTINATION
VALUES ('MED', 'Mediterranean');

INSERT INTO DONOTTOUCH.DESTINATION
VALUES ('ANZ', 'Australia and New Zealand');

INSERT INTO DONOTTOUCH.DESTINATION
VALUES ('AFR', 'Africa');

INSERT INTO DONOTTOUCH.DESTINATION
VALUES ('ASIA', 'Asia');

INSERT INTO DONOTTOUCH.DESTINATION
VALUES ('SA', 'South America');

INSERT INTO DONOTTOUCH.DESTINATION
VALUES ('EU', 'Europe and United Kingdom');

INSERT INTO DONOTTOUCH.DESTINATION
```

```
VALUES ('MEAST', 'Middle East');

INSERT INTO DONOTTOUCH.DESTINATION
VALUES ('SP', 'South Pacific');

INSERT INTO DONOTTOUCH.DESTINATION
VALUES ('NA', 'North America');

INSERT INTO DONOTTOUCH.DESTINATION
VALUES ('OTHR', 'Other');

/* col DESTINATIONCODE for a5 HEADING 'Code'
col DESTINDESC for a30 HEADING 'Description' WORD_WRAPPED

TTITLE 'Destination Codes'
SELECT DESTINATIONCODE, DESTINDESC
FROM DONOTTOUCH.DESTINATION
ORDER BY 1; */

COMMIT;

--Inserting into Class
DELETE FROM DONOTTOUCH.CLASS CASCADE;

INSERT INTO DONOTTOUCH.CLASS
VALUES ('FST', 'First Class');

INSERT INTO DONOTTOUCH.CLASS
VALUES ('BSN', 'Business Class');

INSERT INTO DONOTTOUCH.CLASS
VALUES ('ECN', 'Economy Class');

INSERT INTO DONOTTOUCH.CLASS
VALUES ('OCNV', 'Ocean View');

INSERT INTO DONOTTOUCH.CLASS
VALUES ('INT', 'Interior');

INSERT INTO DONOTTOUCH.CLASS
VALUES ('DLX', 'Delux');

INSERT INTO DONOTTOUCH.CLASS
VALUES ('DBL', 'Double');

INSERT INTO DONOTTOUCH.CLASS
VALUES ('SNG', 'Single');

/* col CLASSCODE for a5 HEADING 'Code'
col CLASSDESC for a30 HEADING 'Description' WORD_WRAPPED

TTITLE '(Trip)Class Types'
```

```
SELECT *
FROM DONOTTOUCH.CLASS
ORDER BY 1; */

COMMIT;

--Inserting into TripType
DELETE FROM DONOTTOUCH.TRIPTYPE;

INSERT INTO DONOTTOUCH.TRIPTYPE
VALUES ('B', 'Business');

INSERT INTO DONOTTOUCH.TRIPTYPE
VALUES ('L', 'Leisure');

INSERT INTO DONOTTOUCH.TRIPTYPE
VALUES ('G', 'Group');

/* col TRIPTYPECODE for a5 HEADING 'Code'
col TRIPDESC for a30 HEADING 'Description' WORD_WRAPPED

TTITLE 'Trip Types'
SELECT *
FROM DONOTTOUCH.TRIPTYPE
ORDER BY 1; */

COMMIT;



--Inserting into RewardCard
DELETE FROM DONOTTOUCH.REWARDCARD CASCADE;

INSERT INTO DONOTTOUCH.REWARDCARD (REWARDCARDDESC)
VALUES ('WestJet Rewards');

INSERT INTO DONOTTOUCH.REWARDCARD (REWARDCARDDESC)
VALUES ('Airmiles');

INSERT INTO DONOTTOUCH.REWARDCARD (REWARDCARDDESC)
VALUES ('SkyMiles');

INSERT INTO DONOTTOUCH.REWARDCARD (REWARDCARDDESC)
VALUES ('Aeroplan');

INSERT INTO DONOTTOUCH.REWARDCARD (REWARDCARDDESC)
VALUES ('MileagePlus');

/* col REWARDCARDDESC for a20 HEADING "Points Program" WORD_WRAPPED
col CARDTYPE for 999 HEADING "No."

TTITLE 'Reward Cards'
```

```
SELECT *
FROM DONOTTOUCH.REWARDCARD
ORDER BY 1;
 */
COMMIT;

--Inserting into PaymentType
DELETE FROM DONOTTOUCH.PAYMENTTYPE CASCADE;

INSERT INTO DONOTTOUCH.PAYMENTTYPE
VALUES (1, 'Full Payment');

INSERT INTO DONOTTOUCH.PAYMENTTYPE
VALUES (2, 'Payment');

INSERT INTO DONOTTOUCH.PAYMENTTYPE
VALUES (3, 'Final Payment');

INSERT INTO DONOTTOUCH.PAYMENTTYPE
VALUES (4, 'Deposit');
/*
col PAYMENTTYPE for 999 HEADING 'No.'
col PAYMENTDESC for a20 HEADING 'Description' WORD_WRAPPED

TTITLE 'Payment Types'
SELECT *
FROM DONOTTOUCH.PAYMENTTYPE
ORDER BY 1; */

COMMIT;

--Inserting into TaxType
DELETE FROM DONOTTOUCH.TAXTYPE CASCADE;

INSERT INTO DONOTTOUCH.TAXTYPE (TAXDESC)
VALUES ('PST Only');

INSERT INTO DONOTTOUCH.TAXTYPE (TAXDESC)
VALUES ('GST Only');

INSERT INTO DONOTTOUCH.TAXTYPE (TAXDESC)
VALUES ('GST + PST');

INSERT INTO DONOTTOUCH.TAXTYPE (TAXDESC)
VALUES ('Environmental Tax');
/*
col TAXTYPE for 999 HEADING 'No.'
col TAXDESC for a20 HEADING 'Description' WORD_WRAPPED

TTITLE 'Tax Types'
SELECT *
FROM DONOTTOUCH.TAXTYPE
```

```
ORDER BY 1; */

COMMIT;
```

Form Scripts

Itinerary/Invoice Script

```
/*Travel Expert Itinerary Script*/
set linesize 90
set pagesize 50
set serveroutput on
set feedback off
set echo off
set verify off

ACCEPT p_itnum number PROMPT 'Enter the itinerary number for the trip: ';

DECLARE
      cursor cursor_cust is
            select c.firstname||' '||c.lastname as name,
                  c.address,
                  c.zip_postal,
                  l.cityname||', '||l.stateprov
            from DONOTTOUCH.customer c join
            DONOTTOUCH.location l on
            c.citycode=l.citycode
            where customerid=(select customerid from DONOTTOUCH.trip where
itinerarynum=&&p_itnum);

      cursor c_travllers is
            select travellername
            from donottouch.triptraveller
            where itinerarynum=&p_itnum;

      cursor c_prod is
            select      tp.startdate,
                        tp.enddate,
                        p.price,
                        p.taxrate
            from donottouch.description d join
            donottouch.tripproduct tp
            on tp.descriptionno=d.descriptionno
            join donottouch.product p
            on p.supplierno=tp.supplierno and p.company=tp.company and
p.productcatno=tp.productcatno
            where tp.itinerarynum=&p_itnum;

      v_dt DATE :=sysdate;
      v_dtc VARCHAR2(100);
      v_cn varchar2(100);
      v_ca varchar2(100);
      v_cp varchar2(10);
      v_cl varchar2(100);

      v_tn varchar2(100);
      v_sdate     date;
      v_edate date;
      v_totalcost number:=0;

      v_psd date;
      v_ped date;
      v_pp number;
      v_ptr number;

      v_totaltax number:=0;
      v_travel donottouch.trip.travellers%TYPE;
      v_custcredit varchar2(100);
      v_credit varchar2(100);

      v_emp varchar2(100);
      v_cnum number;
BEGIN
      select unique(travellers) into v_travel
      from donottouch.trip
      where itinerarynum=&p_itnum;

      v_dtc:=to_char(v_dt,'mm/dd/yyyy');
      dbms_output.put_line('Date: '||v_dtc);
      dbms_output.put_line('Travel Experts Agency');
      dbms_output.put_line('1155 8th Ave S.W.');
      dbms_output.put_line('Calgary, AB. T2P 1N3');
      dbms_output.put_line('Ph: 403-271-9873 Fax: 403-271-9872');
      dbms_output.put_line(chr(10));

      open cursor_cust;
      fetch cursor_cust into v_cn, v_ca, v_cp, v_cl;
      close cursor_cust;

      dbms_output.put_line('************** Invoice **************');
      dbms_output.put_line('To:');
      dbms_output.put_line(v_cn);
      dbms_output.put_line(v_ca);
      dbms_output.put_line(v_cl);
      dbms_output.put_line(v_cp);
      dbms_output.put_line(chr(10));


      select custid into v_cnum
      from donottouch.invoice
      where invoiceno=&&p_inum and itinerarynum=&p_itnum;

      dbms_output.put_line('Customer No.: '||v_cnum);
      select firstname||' '||lastname into v_emp
      from donottouch.employee
      where empid=(select empid
                        from donottouch.customer
                        where customerid=v_cnum);
      dbms_output.put_line('Consultant: '||v_emp||chr(10));
      dbms_output.put_line('Invoice/Itinerary No.: '||&p_itnum);

      dbms_output.put_line('Prepared for: ');

      open c_travllers;
      loop
            fetch c_travllers into v_tn;
            exit when c_travllers%NOTFOUND;
            dbms_output.put_line(chr(9)||v_tn);
      end loop;

      close c_travllers;

      dbms_output.put_line(chr(10));

      open c_prod;
      loop
            fetch c_prod into v_psd, v_ped, v_pp, v_ptr;
            exit when c_prod%NOTFOUND;
            if v_sdate is null then
                  v_sdate :=v_psd;
            elsif v_sdate > v_psd then
                  v_sdate:=v_psd;
            end if;
            if v_edate is null then
                  v_edate :=v_ped;
            elsif v_edate < v_ped then
                  v_edate:=v_ped;
            end if;
            v_totalcost:=v_totalcost+v_pp;
            v_totaltax:=v_totaltax+(v_pp*v_ptr);
      end loop;

      close c_prod;

      dbms_output.put_line('Trip Summary');
      dbms_output.put_line('********************************');
      dbms_output.put_line('Start date: '||v_sdate);
      dbms_output.put_line('End date: '||v_edate);
      dbms_output.put_line('Number of travellers: '||v_travel||chr(10));
      dbms_output.put_line('Subtotal: '||to_char(v_totalcost,
'$99999990.99'));
      dbms_output.put_line('Taxes: '||to_char(v_totaltax,
'$99999990.99')||chr(10));

      select c.creditcardno,cc.creditdesc into v_custcredit, v_credit
      from donottouch.creditcard cc
      join donottouch.custcredit c
      on cc.cardtypeno=c.cardtypeno
      where c.customerid=v_cnum;
      dbms_output.put_line('Billed to:
'||v_credit||chr(9)||v_custcredit||to_char(v_totaltax+v_totalcost,
'$99999990.99'));
END;
/
col 'Supplier' for a10 WORD_WRAPPED
col 'Description' for a30 WORD_WRAPPED
col 'Booking No.' for a10
col 'Start Date' for a10
col 'End Date' for a10
col 'Price' for $999990.99
select p.company as "Supplier",
            d.description as "Description",
            tp.bookingnum as "Booking No.",
            tp.startdate as "Start Date",
            tp.enddate as "End Date",
            p.price as "Price"
from donottouch.description d join
donottouch.tripproduct tp
on tp.descriptionno=d.descriptionno
join donottouch.product p
on p.supplierno=tp.supplierno and p.company=tp.company and
p.productcatno=tp.productcatno
where tp.itinerarynum=&p_itnum
order by 4,6;

DECLARE
      v_dt DATE :=sysdate;
    v_dtc VARCHAR2(100);
BEGIN
      v_dtc:=to_char(v_dt,'mm/dd/yyyy');
      dbms_output.put_line('Date: '||v_dtc);
      dbms_output.put_line('Travel Experts Agency');
      dbms_output.put_line('1155 8th Ave S.W.');
      dbms_output.put_line('Calgary, AB. T2P 1N3');
      dbms_output.put_line('Ph: 403-271-9873 Fax: 403-271-9872');
      dbms_output.put_line(chr(10));
      dbms_output.put_line('For your convience, relevant rewards will be
applied from the following: ');
END;
/
set heading off
col cardno for a20
col 'Name' for a20 WORD_WRAPPED
col rewardcarddesc for a30 word_wrapped
break on 'Name' skip 1

select c.firstname||' '||c.lastname as "Name", r.rewardcarddesc, cr.cardno
from donottouch.customer c
join donottouch.custrewardcard cr
on c.customerid=cr.customerid
join donottouch.rewardcard r
on cr.cardtype=r.cardtype
where c.customerid in (select unique(custid) from donottouch.invoice where
itinerarynum=&p_itnum)
order by 1, 2;


undef p_itnum;
clear col
set heading on
```

Customer Entry Script

```
/*Travel Expert New Customer*/
set linesize 70
set pagesize 50
set serveroutput on
set verify off
set echo off

ACCEPT p_employee number PROMPT 'Enter the Employee ID: ';
ACCEPT p_fn PROMPT 'Enter the first name of the customer: ';
ACCEPT p_ln PROMPT 'Enter the last name of the customer: ';
ACCEPT p_p PROMPT 'Enter the phone number of the customer: ';
ACCEPT p_a PROMPT 'Enter the address of the customer: ';
ACCEPT p_pz PROMPT 'Enter the postal code of the customer: ';
ACCEPT p_cc number PROMPT 'Enter the city code for the customer: ';

ACCEPT p_confirm PROMPT 'Enter YES if the previous entries were correct: ';
DECLARE
      v_loc varchar2(100);
      v_custid number;
BEGIN
      select cityname||', '||stateprov into v_loc
      from donottouch.location
      where citycode=&&p_cc;
      dbms_output.put_line('Travel Experts Agency');
      dbms_output.put_line('1155 8th Ave S.W.');
      dbms_output.put_line('Calgary, AB. T2P 1N3');
      dbms_output.put_line('Ph: 403-271-9873 Fax: 403-271-9872');

      dbms_output.put_line('************** New Customer Details
**************');
      dbms_output.put_line('Customer:'||'&&p_fn'||' '||'&&p_ln');
      dbms_output.put_line('Phone No.: '||'&&p_p');
      dbms_output.put_line('Address: '||'&&p_a');
      dbms_output.put_line(chr(9)||v_loc);
      dbms_output.put_line(chr(9)||'&&p_pz');

      if (lower('&&p_confirm')='yes') then
            insert into DONOTTOUCH.customer
            (empid, firstname, lastname, hphone, address, citycode,
zip_postal)
            values (&&p_employee, '&&p_fn', '&&p_ln', '&&p_p', '&&p_a',
&&p_cc, '&&p_pz');
            commit;
            select customerid into v_custid
            from donottouch.customer
            where empid=&&p_employee and firstname='&&p_fn'
            and lastname='&&p_ln' and address='&&p_a';
            dbms_output.put_line('Customer ID: '||v_custid);
      else
            dbms_output.put_line('Cancelling Form');
      end if;

END;
/


undef p_employee;
undef p_fn;
undef p_ln;
undef p_p;
undef p_a;
undef p_cc;
undef p_pz;
```

Commissions Script

```
/*Travel Expert Commission Script*/
set linesize 70
set pagesize 50
set serveroutput on
set feedback off
set echo off
set verify off

ACCEPT p_employee number PROMPT 'Enter the Employee number that you
wish to print commission: ';
DECLARE
      cursor cursor_products is
            select c.listno, c.commissionamount, t.commissiontype
            from DONOTTOUCH.commission c join
            DONOTTOUCH.commissiontype t
            on t.commissionno = c.commissionno
            where c.empid=&p_employee
            order by 1, 3, 2;

      cursor cursor_emp is
            select firstname||' '||lastname as name,
                  phone,
                  email
            from DONOTTOUCH.employee
            where empid=&p_employee;

      v_dt DATE :=sysdate;
    v_dtc VARCHAR2(100);

      v_ln number;
      v_ca number;
      v_ct varchar2(100);

      v_en varchar2(100);
      v_ep varchar2(20);
      v_ee varchar2(100);
BEGIN
      v_dtc:=to_char(v_dt,'mm/dd/yyyy');
      dbms_output.put_line('Travel Experts Agency');
      dbms_output.put_line('1155 8th Ave S.W.');
      dbms_output.put_line('Calgary, AB. T2P 1N3');
      dbms_output.put_line('Ph: 403-271-9873 Fax: 403-271-9872');
      open cursor_emp;
      fetch cursor_emp into v_en, v_ep, v_ee;
      close cursor_emp;
      dbms_output.put_line('************** Commissions
**************');
      dbms_output.put_line('Report Date: ' || v_dtc);
      dbms_output.put_line('Employee ID:' || &p_employee);
      dbms_output.put_line('Employee:' || v_en);
      dbms_output.put_line('Employee Email:' || v_ee);
      dbms_output.put_line('Employee Phone:' || v_ep);

      dbms_output.put_line(chr(10));
      dbms_output.put_line('List#'||chr(9)||chr(9)||'Amount'||chr(9)||c
hr(9)||'Status');
      dbms_output.put_line('*****************************************')
;
      open cursor_products;
      loop
           fetch cursor_products into v_ln, v_ca, v_ct;
           exit when cursor_products%NOTFOUND;


      dbms_output.put_line(to_char(v_ln,'999999')||chr(9)||to_char(v_ca
,'$999999.99')||chr(9)||chr(9)||v_ct);
      end loop;

      close cursor_products;
END;
/

undef p_employee;
```

Appendix E – Extra Scripts

Sample Database Verification Script

```
SET Linesize 40
SET pagesize 15
set echo off
show user;
SET feedback off

SET long 1000
TTITLE 'Verify'


SET HEADING OFF

SELECT 'Instance name: ',instance_name
FROM v$instance
UNION
SELECT 'Database name: ', name
FROM v$database;

TTITLE OFF
BTITLE OFF


--need to add tablespace location?

SELECT 'Date: ', sysdate
FROM dual;
show user

SET linesize 70
SET pagesize 30
SET heading ON
COL name FOR a50 WORD_WRAPPED
TTITLE 'Files'

SELECT status, name
FROM v$datafile
UNION
SELECT 'Control', name
FROM v$controlfile
UNION
SELECT status, member
FROM v$logfile
ORDER BY 1 desc;

TTITLE 'Log Files'

SET pagesize 40
COL name for a10 WORD_WRAPPED
COL value for a50 WORD_WRAPPED

SELECT name, value
FROM v$diag_info
ORDER BY 2;

SELECT group#, status, members
FROM v$log
ORDER BY 1,2;

SET pagesize 80
COL 'Loc' FOR a52 WORD_WRAPPED
COL 'TS Name' FOR a10 WORD_WRAPPED
COL ts# for 999

BREAK ON ts# ON name

TTITLE 'TableSpaces'

SELECT d.ts#, d.name "Loc", t.name "TS Name"
FROM v$datafile d
JOIN v$tablespace t
ON d.ts# = t.ts#
ORDER BY 1;

COL tablespace_name FOR a20 HEADING "TS Name"

SELECT tablespace_name, status, logging, retention, block_size "Block
Size"
FROM dba_tablespaces
ORDER BY 1;

SET linesize 40
SET pagesize 15
COL value for 9999999999999
COL name for a25 WORD_WRAPPED
TTITLE 'SGA'

SELECT NAME, value
FROM v$sga;

TTITLE 'Parameters'
COL name for a30 WORD_WRAPPED
COL value for a35
SET linesize 70
SET pagesize 20

SELECT NAME, value
FROM v$parameter
WHERE lower(name) = 'compatible'
OR lower(name) = 'instance_name'
OR lower(name) = 'db_name'
OR lower(name) = 'diagnostic_dest'
OR lower(name) = 'sga_target'
OR lower(name) LIKE 'db_recovery_file%'
OR lower(name) = 'processes'
OR lower(name) IN ('undo_tablespace', 'undo_management',
'undo_retention', 'db_block_size')
ORDER BY 1;


CLEAR COL
CLEAR BREAK

SET FEEDBACK ON
set echo on
TTITLE OFF
BTITLE OFF
SET linesize 60
SET pagesize 60
SET long 80
```

Data Warehouse Scripts

SQL LDR

```
--commands used in the command line
sqlldr thedba/Oracle control=c:\TEA\loadcust.ctl log=c:\TEA\cust.log
sqlldr thedba/Oracle control=c:\TEA\loadsale.ctl log=c:\TEA\sale.log
sqlldr thedba/Oracle control=c:\TEA\loadsupp.ctl log=c:\TEA\supp.log
```

```
--loadsupp.ctl
OPTIONS(skip=1, direct=y)
LOAD DATA
      INFILE 'c:\tea\projectdata_supp.csv'
      BADFILE 'c:\tea\bad_supp.csv'
      DISCARDFILE 'c:\tea\discard.csv'
      TRUNCATE INTO TABLE DONOTTOUCH.supp_adhoc
      Fields terminated by "," optionally enclosed '"'
      (
            prod_supp integer external
            ,prod_cat integer external
            ,supp_off integer external
            ,prod_desc char
            ,cont_name char
            ,company char
            ,address char
            ,address2 char
            ,city char
            ,prov char
            ,postal char
            ,country char
            ,phone char
            ,fax char
            ,email char
            ,web char
            ,represents char
            ,affiliation char
      )

--loadcust.ctl
OPTIONS(skip=1, direct=y)
LOAD DATA
      INFILE 'c:\tea\projectdata_cust.csv'
      BADFILE 'c:\tea\bad_cust.csv'
      DISCARDFILE 'c:\tea\discard_supp.csv'
      TRUNCATE INTO TABLE DONOTTOUCH.cust_adhoc
      Fields terminated by "," optionally enclosed by '"'
      (
            cust_id integer external
            ,first_name char
            ,last_name char
            ,agent char
            ,email char
            ,home_phone char
            ,business_phone char
            ,birthdate date 'YYYY-MM-DD'
            ,address char
            ,city char
            ,postal char
            ,province char
            ,country char
      )

--loadsale.ctl
OPTIONS(skip=1, direct=y)
LOAD DATA
      INFILE 'c:\tea\projectdata_sale.csv'
      BADFILE 'c:\tea\bad_sale.csv'
      DISCARDFILE 'c:\tea\discard.csv'
      TRUNCATE INTO TABLE DONOTTOUCH.sale_adhoc
      Fields terminated by "," optionally enclosed '"'
      (
      sales_date date 'YYYY-MM-DD'
      ,cust_id integer external
      ,itinerary char
      ,agent char
      ,booking char
      ,prod_cat integer external
      ,supplier_id integer external
      ,supp_off integer external
      ,trip_start date 'YYYY-MM-DD'
      ,trip_end date 'YYYY-MM-DD'
      ,class char
      ,num_travellers integer external
      ,product char
      ,description char
      ,destination char
      ,dest_id char
      ,credit_card char
      ,expiry date 'YYYY-MM-DD'
      ,card_num char
      ,bill_date date 'YYYY-MM-DD'
      ,bill_desc char
      ,base_price float external
      ,tot_price float external
      ,bill_amt float external
      ,agency_fee_code char
      ,agency_fee_amt float external
      ,agency_comm float external
      )
```

Creating Fact Table

```
CREATE SEQUENCE donottouch.factseq
      INCREMENT BY 1
      START WITH 1
      NOCYCLE
      NOCACHE;

CREATE TABLE donottouch.fact
(
sequencenum number(10) default donottouch.factseq.nextval,
destination varchar2(100) default null,
cust_name varchar2(100) default null,
emp_name varchar2(100) default null,
year number(4) default null,
month number(2) default null,
day number(2) default null,
supplier number(10) default null,
product varchar2(30) default null,
fee_type varchar2(50) default null,
class varchar2(100) default null,
commission varchar2(25) default null,
amount number(10,2) default 0,
quantity number(10) default 1,
bphone varchar2(15) default null,
hphone varchar2(15) default null,
address varchar2(100) default null,
zipcode varchar2(10) default null,
city varchar2(50) default null,
province varchar2(20) default null,
country varchar2(20) default null,
contact varchar2(50) default null
)
partition by hash (sequencenum)
partitions 2
store in (userdata01,userdata02);
```

--Tuning through create dimension

```
create dimension speedup_fact
level yy is fact.year
level mm is fact.month
level dd is fact.day
level country is fact.country
level prov is fact.province
level city is fact.city
level address is fact.address
level zip is fact.zipcode
hierarchy ymd
(dd child of mm child of yy)
hierarchy adrs
(zipcode child of address)
hierarchy loc
(city child of prov child of country);

--LOADING FACT TABLE
merge into donottouch.FACT x
using
(
select d.destindesc as destination,
      to_char(tp.startdate,'yyyy') year,
      to_char(tp.startdate,'mm') month,
      to_char(tp.startdate,'dd') day,
      tp.supplierno as supplier,
      productcategory as product,
      c.classdesc as class,
      1 as quantity,
      p.price as amount,
      s.phone,
      s.address,
      s.zip_postal,
      loc.cityname,
      loc.stateprov,
      loc.countryname,
      tp.listno
from destination d join tripproduct tp on d.destinationcode =
tp.destinationcode
join class c on c.classcode=tp.classcode
join product p on p.productcatno=tp.productcatno and
tp.supplierno=p.supplierno
join productcat pc on pc.productcatno=p.productcatno
join supplier s on s.supplierno=tp.supplierno and s.companyname=tp.company
join location loc on loc.citycode=s.citycode
)y
on (y.listno = x.listno)
when matched then update set
destination=y.destination,year=y.year,month=y.month,day=y.day,supplier=y.supp
lier,product=y.product,class=y.class,quantity=y.quantity,
amount=y.amount,bphone=y.phone,address=y.address,zipcode=y.zip_postal,city=y.
cityname,province=y.stateprov,country=y.countryname
when not matched then insert (destination, year, month, day,
supplier,product,class,quantity,amount,bphone, address, zipcode, city,
province, country)
```

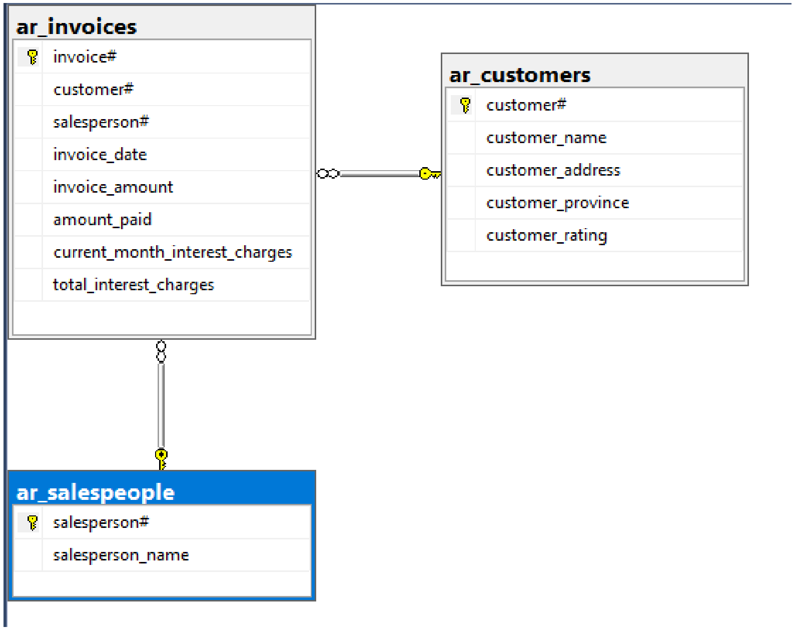
```
      values(y.destination,
y.year,y.month,y.day,y.supplier,y.product,y.class,y.quantity,y.amount,y.phone
, y.address,y.zip_postal, y.cityname,y.stateprov, y.countryname);

merge into donottouch.fact x
using
(
select ct.commissiontype as commission,
      e.firstname||' '||e.lastname as name,
      c.commissionamount as amount,
      e.phone,
      to_char(tp.startdate,'yyyy') year,
      to_char(tp.startdate,'mm') month,
      tp.listno
from tripproduct tp join commission c
on tp.listno=c.listno
join employee e on e.empid=c.empid
join commissiontype ct on ct.commissionno=c.commissionno
) y
on (y.listno = x.listno and y.commission=x.commission)
when matched then update set emp_name=y.name, bphone=y.phone, quantity=1,
amount=y.amount, year=y.year, month=y.month
when not matched then insert (commission,year,month, emp_name,
bphone,quantity, listno, amount) values(y.commission,y.year,y.month, y.name,
y.phone, 1, y.listno, y.amount);

merge into donottouch.fact x
using
(select c.firstname||' '||c.lastname as name,
      c.hphone,
      c.bphone,
      c.address,
      c.zip_postal,
      l.cityname,
      l.stateprov,
      l.countryname,
      to_char(i.billingdate,'yyyy') as year,
      to_char(i.billingdate,'mm') as month,
      i.billamount,
      i.itinerarynum||i.invoiceno as invoice
from customer c join invoice i on i.custid=c.customerid
join location l on l.citycode = c.citycode
) y
on (y.invoice=x.invoiceno)
when matched then update set cust_name=y.name,
hphone=y.hphone,bphone=y.bphone,address=y.address,zipcode=y.zip_postal,
city=y.cityname,province=y.stateprov,country=y.countryname,year=y.year,month=
y.month,amount=y.billamount
when not matched then insert (cust_name, hphone,bphone,address,zipcode, city,
province,country,year, month, amount, quantity, invoiceno)
      values(y.name, y.hphone,y.bphone,y.address,y.zip_postal, y.cityname,
y.stateprov,y.countryname,y.year, y.month, y.billamount, 1, y.invoice);
```

```
merge into donottouch.fact x
using
(select f.feetypedesc,
      f.feeamount,
      i.itinerarynum||i.invoiceno as invoice
from invoice i join feetype f
on i.feecode = f.feecode) y
on (y.invoice = x.invoiceno and x.fee_type = y.feetypedesc)
when matched then update set amount=y.feeamount
when not matched then insert (quantity,amount,fee_type,invoiceno)
values(1, y.feeamount, y.feetypedesc, y.invoice);

--Dimensions
create table donottouch.destinatino_tot(destination varchar2(100) primary
key, quantity number(10));
create table donottouch.class_tot(class varchar2(100) primary key, quantity
number(10),total number(10,2));
create table donottouch.fee_tot(fee_type varchar2(50) primary key, total
number(10,2));
create table donottouch.comm_tot(commission varchar2(25) primary key,
quantity number(10), total number (10,2));
create table donottouch.trips_year(year number(4) primary key, quantity
number(10));
create table donottouch.trips_month(year number(4), month number(2), quantity
number(10), primary key (year,month));
create table donottouch.trips_day(year number(4), month number(2), day
number(2), quantity number(10), primary key (year, month, day));
create table donottouch.emp_comm(name varchar2(100), commission varchar2(25),
quantity number(10), total number(10,2), primary key (name,commission));
create table donottouch.cust_tot(name varchar2(100) primary key, total
number(10,2));
create table donottouch.cust_tot_year(name varchar2(100), year number(4),
total number(10,2), primary key (name, year));
create table donottouch.cust_tot_month(name varchar2(100), year number(4),
month number(2), total number(10,2), primary key (name, year, month));
create table donottouch.prod_sold(supplier number(10), product varchar2(50),
quantity number(10), primary key (supplier, product));
create table donottouch.supp_tot(supplier number(10) primary key, quantity
number(10), total number(10,2));
create table donottouch.prod_tot(product varchar2(50) primary key, quantity
number(10));

--LOADING DIMENSIONS
--returns the number of trips taken to particular destination group
merge into donottouch.destinatino_tot x
using
(select destination, count(destination) as quantity
from donottouch.fact
group by destination
having destination is not null) y
on (y.destination = x.destination)
when matched then update set quantity=y.quantity
when not matched then insert values(y.destination, y.quantity);
```

```
--returns the number and total spent for trips of certain class
merge into donottouch.class_tot x
using
(
select class,count(class) as quantity, sum(amount) as total
from fact
group by class
having class is not null
) y
on (y.class = x.class)
when matched then update set quantity=y.quantity,total=y.total
when not matched then insert values(y.class,y.quantity, y.total);

--returns the total obtained through different fees
merge into donottouch.fee_tot x
using
(select fee_type as fee, amount*count(fee_type) as total
from fact
group by fee_type,amount
having fee_type is not null
) y
on (y.fee = x.fee_type)
when matched then update set total=y.total
when not matched then insert values(y.fee, y.total);
--returns total count and amount of different commissions
merge into donottouch.comm_tot x
using
(
select commission, count(quantity) as quantity, sum(amount) as total
from fact
group by commission
having commission is not null
) y
on (y.commission = x.commission)
when matched then update set total=y.total,quantity=y.quantity
when not matched then insert values(y.commission,y.quantity, y.total);
--returns the count of all the trips
merge into donottouch.trips_year x
using
(select year, count(quantity) as quantity
from (select listno, year, quantity from fact where listno is not null and
year is not null) z
group by year) y
on (y.year = x.year)
when matched then update set quantity=y.quantity
when not matched then insert values(y.year, y.quantity);
--month
merge into donottouch.trips_month x
using
(select year, month,count(quantity) as quantity
from (select listno, year,month, quantity from fact where listno is not null
and year is not null and month is not null) z
group by year,month) y
```

```
on (y.year = x.year and y.month=x.month)
when matched then update set quantity=y.quantity
when not matched then insert values(y.year, y.month, y.quantity);
--day
merge into donottouch.trips_day x
using
(select year, month,day,count(quantity) as quantity
from (select listno, year,month,day, quantity from fact where listno is not
null and day is not null) z
group by year,month,day) y
on (y.year = x.year and y.month=x.month and y.day=x.day)
when matched then update set quantity=y.quantity
when not matched then insert values(y.year, y.month,y.day, y.quantity);
--employee commissions total and amount divided into the types of
commmissions
merge into donottouch.emp_comm x
using
(
select emp_name,commission, count(quantity) as quantity, sum(amount) as total
from fact
group by emp_name,commission
having emp_name is not null and commission is not null
) y
on (y.emp_name = x.name)
when matched then update set total=y.total,quantity=y.quantity
when not matched then insert values(y.emp_name,y.commission,y.quantity,
y.total);
--customers total spent
merge into donottouch.cust_tot x
using
(
select cust_name as name, sum(amount) as total
from fact
group by cust_name
having cust_name is not null) y
on (y.name = x.name)
when matched then update set total=y.total
when not matched then insert values(y.name, y.total);
--customer total spent - year
merge into donottouch.cust_tot_year x
using
(select cust_name as name, year,sum(amount) as total
from fact
group by cust_name, year
having cust_name is not null and year is not null
order by 1,2) y
on (y.name = x.name and y.year=x.year)
when matched then update set total=y.total
when not matched then insert values(y.name,y.year, y.total);
--customer total spent - month
merge into donottouch.cust_tot_month x
using
(select cust_name as name, year,month,sum(amount) as total
from fact
group by cust_name, year,month
having cust_name is not null and year is not null and month is not null
order by 1,2,3) y
on (y.name = x.name and y.year=x.year and x.month=y.month)
when matched then update set total=y.total
when not matched then insert values(y.name,y.year,y.month, y.total);
--number of times a particular product was sold
merge into donottouch.prod_sold x
using
(
select supplier, product, count(quantity) as total
from fact
group by supplier,product
having supplier is not null and product is not null
) y
on (y.supplier = x.supplier and y.product = x.product)
when matched then update set quantity=y.total
when not matched then insert values(y.supplier,y.product, y.total);
--number of times a particular group of product was sold
merge into donottouch.prod_tot x
using
(
select product, count(quantity) as total
from fact
group by product
having product is not null
order by 1
) y
on (y.product = x.product)
when matched then update set quantity=y.total
when not matched then insert values(y.product, y.total);
--number and total of things a supplier sold something
merge into donottouch.supp_tot x
using
(select supplier, count(quantity) as quantity, sum(amount) as total
from fact
group by supplier
having supplier is not null) y
on (y.supplier = x.supplier)
when matched then update set quantity=y.quantity,total= y.total
when not matched then insert values(y.supplier, y.quantity, y.total);


create or replace procedure emp_commissions
as
      cursor com is select emp_name, commission, amount
                  from fact
                  where emp_name is not null and commission is not null
                  order by emp_name desc;
      com_v com%rowtype;
      total number(10,2):=0;
      name varchar2(50);
      status varchar2(50);
begin
      open com;
      fetch com into com_v;

      dbms_output.put_line('Employee'||chr(9)||'Commission
Status'||chr(9)||chr(9)||'Amount');
      loop
            exit when com%notfound;
            if com_v.emp_name != name then

      dbms_output.put_line(chr(10)||chr(9)||chr(9)||chr(9)||chr(9)||'Total:
$'||total||chr(10));
                  total:=0;
                  name :=com_v.emp_name;
                  status :=com_v.commission;

      dbms_output.put_line(name||chr(9)||chr(9)||status||chr(9)||chr(9)||com_
v.amount);
                  total:=total+com_v.amount;
            elsif com_v.commission!=status then
                  status:=com_v.commission;

      dbms_output.put_line(chr(9)||chr(9)||status||chr(9)||chr(9)||com_v.amou
nt);
                  total:=total+com_v.amount;
            elsif com_v.commission = status then

      dbms_output.put_line(chr(9)||chr(9)||chr(9)||chr(9)||chr(9)||chr(9)||co
m_v.amount);
                  total:=total+com_v.amount;
            elsif name is null then
                  name :=com_v.emp_name;
                  status :=com_v.commission;

      dbms_output.put_line(name||chr(9)||status||chr(9)||chr(9)||com_v.amount
);
                  total:=total+com_v.amount;
            end if;
            fetch com into com_v;
      end loop;
      dbms_output.put_line(chr(10)||chr(9)||chr(9)||chr(9)||chr(9)||'Total:
$'||total||chr(10));
      close com;
end;
/
exec emp_commissions
```

Example Database Locking Script



```
create table usersecurity(
username varchar2(30) primary key,
securitylevel numeric(2));

insert into usersecurity values (user, 0);
commit;

create table ar_sales_log
(salesp# number(3) primary key,
security number(2),
createuser varchar2(30),
createdate date,
createloc varchar2(30),
updateuser varchar2(30),
updatedate date,
updateloc varchar2(30));

create table ar_inv_log
(inv# number(3) primary key,
security number(2),
createuser varchar2(30),
createdate date,
createloc varchar2(30),
updateuser varchar2(30),
updatedate date,
updateloc varchar2(30));

create table ar_cust_log
(cust# number(3) primary key,
security number(2),
createuser varchar2(30),
createdate date,
createloc varchar2(30),
updateuser varchar2(30),
updatedate date,
updateloc varchar2(30));

insert into ar_cust_log select customer#, 0,user, sysdate,
userenv('terminal'), null,null,null from ar_customers;
insert into ar_inv_log select invoice#, 0,user, sysdate,
userenv('terminal'), null,null,null from ar_invoices;
insert into ar_sales_log select salesperson#, 0,user, sysdate,
userenv('terminal'), null,null,null from ar_salesperson;
commit;

create view ar_cust_data as
select customer#, customer_name, customer_address, customer_province,
customer_rating
from ar_customers a
join ar_cust_log b on a.customer#=b.cust#
where security<=(select securitylevel from usersecurity where
username=user);


create view ar_sales_data as
select salesperson#, salesperson_name
from ar_salespeople a
join ar_sales_log b on a.salesperson#=b.salesp#
where security<=(select securitylevel from usersecurity where
username=user);

create view ar_inv_data as
select invoice#,
     customer#,
     salesperson#,
     invoice_date,
     invoice_amount,
     amount_paid,
     current_month_interest_charges,
     total_interest_charges
from ar_invoices a
join ar_inv_log b on a.invoice#=b.inv#
where security<=(select securitylevel from usersecurity where
username=user);

create table ar_cust_log_det
(
cust# number(3),
logdate date,
datatype char(1) check (datatype in ('S', 'I', 'U', 'D')),
loguser varchar2(30),
primary key(cust#, logdate)
);
```

```
create table ar_inv_log_det
(
inv# number(3),
logdate date,
datatype char(1) check (datatype in ('S', 'I', 'U', 'D')),
loguser varchar2(30),
primary key(inv#, logdate)
);

create table ar_sales_log_det
(
salesp# number(3),
logdate date,
datatype char(1) check (datatype in ('S', 'I', 'U', 'D')),
loguser varchar2(30),
primary key(salesp#, logdate)
);

--Trigger and procedures for Customer
create or replace trigger ar_cust_watch
after insert or update or delete on ar_customers
for each row
begin
if inserting then
      insert into ar_cust_log values (:new.customer#, 0, user, sysdate,
userenv('terminal'), null, null,null);
      insert into ar_cust_log_det values(:new.customer#, sysdate, 'I',
user);
elsif updating  then
      update ar_cust_log set updateuser=user, updatedate=sysdate,
updateloc=userenv('terminal') where cust#=:old.customer#;
      insert into ar_cust_log_det values (:new.customer#, sysdate, 'U',
user);
elsif deleting then
      insert into ar_cust_log_det values (:new.customer#, sysdate, 'D',
user);
end if;
end;
/

--delete customer
create or replace procedure ar_cust_del (cust# number)
as
begin
delete from ar_customers
where customer# = cust#;
end;
/
--insert one record
```

```
create or replace procedure ar_cust_ins (cust# number, name varchar2,
addr varchar2, prov varchar2, rate number)
as
begin
insert into ar_customers (customer#, customer_name, customer_address,
customer_province, customer_rating)
values (cust#, name, addr, prov, rate);
end;
/
--update
create or replace procedure ar_cust_upd (cust# number, name varchar2,
addr varchar2, prov varchar2, rate number)
as
begin
update ar_customers
set customer_name=name,
      customer_address=addr,
      customer_province=prov,
      customer_rating=rate
      where customer# = cust#;
end;
/
--select 1
create or replace procedure ar_cust_get (cust# in number,cust out
number, cname out varchar2, addr out varchar2, prov out varchar2, rate
out number)
as
begin
      select customer#, customer_name, customer_address,
customer_province, customer_rating
      into cust, cname, addr, prov, rate
      from ar_customers
      where customer#=cust#;
      insert into ar_cust_log_det values (cust#, sysdate, 'S', user);
end;
/
--select all
create or replace procedure ar_cust_geta (custdata out sys_refcursor)
as
begin
for x in (select * from ar_customers) loop
      insert into ar_cust_log_det
      values (x.customer#, sysdate, 'S', user);
end loop;
open custdata for select * from ar_customers;
end;
/

--Trigger and procedures for invoices
create or replace trigger ar_inv_watch
after insert or update or delete on ar_invoices
for each row
```

```
begin
if inserting then
      insert into ar_inv_log values (:new.invoice#, 0, user, sysdate,
userenv('terminal'), null, null,null);
      insert into ar_inv_log_det values(:new.invoice#, sysdate, 'I',
user);
elsif updating  then
      update ar_inv_log set updateuser=user, updatedate=sysdate,
updateloc=userenv('terminal') where inv#=:old.invoice#;
      insert into ar_inv_log_det values (:new.invoice#, sysdate, 'U',
user);
elsif deleting then
      insert into ar_inv_log_det values (:new.invoice#, sysdate, 'D',
user);
end if;
end;
/

--delete
create or replace procedure ar_inv_del (inv# number)
as
begin
delete from ar_invoices
where invoice# = inv#;
end;
/
--insert one record
create or replace procedure ar_inv_ins (inv# number, cust# number,
salesp# number, indate date, inamt number, amtp number, curint number,
totint number)
as
begin
insert into ar_invoices (invoice#, customer#, salesperson#,
invoice_date, invoice_amount, amount_paid,
current_month_interest_charges, total_interest_charges)
values (inv#, cust#, salesp#, indate, inamt, amtp, curint, totint);
end;
/
--update
create or replace procedure ar_inv_upd (inv# number, cust# number,
salesp# number, indate date, inamt number, amtp number, curint number,
totint number)
as
begin
update ar_invoices
set customer#=cust#,
      salesperson#=salesp#,
      invoice_date=indate,
      invoice_amount=inamt,
      amount_paid=amtp,
      current_month_interest_charges=curint,
      total_interest_charges=totint
      where invoice# = inv#;
end;
/
--select 1
create or replace procedure ar_inv_get (inv# in number,inv out number,
cust# out number, salesp# out number, indate out date, inamt out
number, amtp out number, curint out number, totint out number)
as
begin
      select invoice#,
            customer#,
            salesperson#,
            invoice_date,
            invoice_amount,
            amount_paid,
            current_month_interest_charges,
            total_interest_charges
            into inv,cust#,salesp#,indate,inamt, amtp,curint,totint
      from ar_invoices
      where invoice#=inv#;
      insert into ar_inv_log_det values (inv#, sysdate, 'S', user);
end;
/
--select all
create or replace procedure  ar_inv_geta (datax out sys_refcursor)
as
begin
for x in (select * from ar_invoices) loop
      insert into ar_inv_log_det
      values (x.invoice#, sysdate, 'S', user);
end loop;
open datax for select * from ar_invoices;
close datax;
end;
/

--Trigger and procedures for salespeople
create or replace trigger ar_sales_watch
after insert or update or delete on ar_salespeople
for each row
begin
if inserting then
      insert into ar_sales_log values (:new.salesperson#, 0, user,
sysdate, userenv('terminal'), null, null,null);
      insert into ar_sales_log_det values(:new.salesperson#, sysdate,
'I', user);
elsif updating  then
      update ar_sales_log set updateuser=user, updatedate=sysdate,
updateloc=userenv('terminal') where salesp#=:old.salesperson#;
      insert into ar_sales_log_det values (:new.salesperson#, sysdate,
'U', user);
elsif deleting then
```

```
     insert into ar_sales_log_det values (:new.salesperson#, sysdate,
'D', user);
end if;
end;
/

--delete customer
create or replace procedure ar_sales_del (salesp# number)
as
begin
delete from ar_salespeople
where salesperson# = salesp#;
end;
/
--insert one record
create or replace procedure ar_cust_ins (salesp# number, name
varchar2)
as
begin
insert into ar_salespeople (salesperson#, salesperson_name)
values (salesp#, name);
end;
/
--update
create or replace procedure ar_sales_upd (salesp# number, name
varchar2)
as
begin
update ar_salespeople
set salesperson_name=name
     where salesperson# = salesp#;
end;
/
--select 1
create or replace procedure ar_sales_get (salesp# in number, sales out
number, name out varchar2)
as
begin
     select salesperson#, salesperson_name into sales, name
     from ar_salespeople
     where salesperson#=salesp#;

     insert into ar_sales_log_det values (salesp#, sysdate, 'S',
user);
end;
/
--select all
create or replace procedure ar_sales_geta (datax out sys_refcursor)
as
begin
for x in (select * from ar_salespeople) loop
     insert into ar_sales_log_det
     values (x.salesperson#, sysdate, 'S', user);
end loop;
open datax for select * from ar_salespeople;
end;
/
```

Linux Scripting

Datapump Script

```
#!usr/bim/ksh

#call my functions in...
. /home/oracle/Scripts/functions.ksh

if [ $# != 1 ]; then
     echo "Needs to  be called with 1 parameter"
     exit 1
fi

#
#Validate parameter is a valid instance
#

ORACLE_SID=$1
dbhome > /dev/null

if [ $? != 0 ]; then
     echo "${ORACLE_SID} is not a valid instance in `hostname`"

fi

echo "Oracle Sid is good"

ORAENV_ASK=NO
. oraenv >/dev/null 2>&1

if [ $? != 0 ]; then

     echo "Unable to set environment for $ORACLE_SID"
     exit 1
fi

echo "Good to continue"

YYMMDD=`date +'%Y%m%d%H%M'`
LOGFILE=/home/oracle/Logs/oradp_${ORACLE_SID}_${YYMMDD}.out
echo_msg "Script Started"
BU_LOCATION=/u03/BACKUP


#
# ensure backup location is good
```

```
#

if [ ! -d ${BU_LOCATION}/${ORACLE_SID} ]; then
     mkdir ${BU_LOCATION}/${ORACLE_SID} > /dev/null 2>&1
     if [ $? != 0 ]; then
          echo_msg "Failed to create backup location :
${BU_LOCATION}/${ORACLE_SID}"
          exit 1
     fi
fi

sqlplus -s /nolog << EOF
whenever sqlerror exit 1
connect / as sysdba
create or replace directory ORADP as '${BU_LOCATION}/${ORACLE_SID}';
exit
EOF

if [ $? != 0 ]; then
     echo_msg "Unable to create ORADP as
'${BU_LOCATION}/${ORACLE_SID}'"
     exit 1
fi

echo_msg "expdp running............."
DUMPFILE=$ORACLE_SID_${YYMMDD}.dp
DPLOGFILE=$ORACLE_SID_${YYMMDD}.log

expdp \"/ as sysdba \"  directory=ORADP schemas=SYSTEM
dumpfile={DUMPFILE} logfile={DPLOGFILE} >/dev/null 2>&1

if [ $? != 0 ]; then
     echo_msg "expdp reported failures"
     cat ${BU_LOCATION}/${ORACLE_SID} | tee -a ${LOGFILE}
     exit 1
else
     echo_msg "expdp completed with no error"
     cat  ${BU_LOCATION}/${ORACLE_SID} | tee -a ${LOGFILE}

     cd ${BU_LOCATION}/${ORACLE_SID}
     echo_msg "Zipping ${DUMPFILE}"
     gzip ${DUMPFILE}
     if [ $! != 0 ]; then


     fi
fi

echo_msg "Script Ended"
```

Hot Backup Script

```
#!/usr/bin/ksh
#
#Created by: Sally
#
# Assumptions about the database the script expects:
#    database to be open in read write
#    will change the database to archive log mode if not
#    connect / as sysdba
#    network files to be in default or in /etc
#    running with spfile
#
# Script improvement considerations:
#    add DBVERIFY
#    make backup locations as parameter
#    restore pfile a location....
#Grab functions for echo_msg, echo_rec, echo_rec_msg from:
. ${HOME}/Scripts/functions.ksh

if [ $# != 1 ]; then
   echo "orahot.sh needs to be called with 1 parameter - Instance
name"
   exit 1
fi

#
# Validate parameter is a valid instance
#

ORACLE_SID=$1
dbhome > /dev/null 2>&1

if [ $? != 0 ]; then

   echo "${ORACLE_SID} is not a valid instance on `hostname`"
   exit 1
fi

ORAENV_ASK=NO
. oraenv > /dev/null 2>&1

if [ $? != 0 ]; then

   echo "Unable to set environment for $ORACLE_SID"
   exit 1
fi

#variables...
```

```
YYMMDD=`date +'%Y%m%d%H%M'`
LOGFILE=${HOME}/Logs/orahot_${ORACLE_SID}_${YYMMDD}.out
BU_LOCATION=/u05/BACKUP

BU_LOC_FILE=${BU_LOCATION}/${ORACLE_SID}/orahot_${YYMMDD}
REC_FILE=${BU_LOC_FILE}/orahot_recovery_${ORACLE_SID}.sh

echo_msg "Script Started"

if [ ! -d ${BU_LOCATION}/${ORACLE_SID} ]; then

  mkdir ${BU_LOCATION}/${ORACLE_SID} > /dev/null 2>&1
  if [ $? != 0 ]; then
    echo_msg "Failed to create backup location :
${BU_LOCATION}/${ORACLE_SID}"
    exit 1
  fi

fi

mkdir ${BU_LOC_FILE} > /dev/null 2>&1
if [ $? != 0 ]; then
      echo_msg "Failed to create final backup location: ${BU_LOC_FILE}"
      exit 2
fi

TMPFILE=${HOME}/Logs/orahottemp_${ORACLE_SID}_${YYMMDD}.tmp

#Checking database status

sqlplus -s / as sysdba << EOF > /dev/null 2>&1
whenever sqlerror exit 1;
set echo off
set heading off
set feedback off
set trimspool on
set pagesize 0
set linesize 2048
spool ${TMPFILE}
select open_mode from v\$database;
spool off
exit
EOF


if [ $? != 0 ]; then
      echo_msg "Error checking database status."
      exit 2
```

```
fi


echo_msg "Temp. File: ${TMPFILE}"
FILEOUT=`cat ${TMPFILE}`

if [ "${FILEOUT}"=="READ WRITE" ]; then
      echo_msg "Database status: ${FILEOUT}"
else
      echo_msg "Database status: ${FILEOUT}"
      echo_msg "Check database status before rerunning this script"
      exit 1
fi

#Checking db archive log status
sqlplus -s /nolog << EOF > /dev/null 2>&1
whenever sqlerror exit 1;
connect / as sysdba
set echo off
set heading off
set feedback off
set trimspool on
set pagesize 0
set linesize 2048
spool ${TMPFILE}
select log_mode from v\$database;
spool off
exit
EOF


if [ $? != 0 ]; then
      echo_msg "Error checking log status database."
      exit 2
fi

echo_msg "Logfile: ${LOGFILE}"
echo_msg "Recovery File: ${REC_FILE}"
echo_msg "Backup location: ${BU_LOC_FILE}"
ARMODE=`cat ${TMPFILE}`
if [ "${ARMODE}"=="ARCHIVELOG" ]; then
      echo_msg "Database archive mode confirmed"
else
      echo_msg "DB not in archive log mode"
      exit 1
#if not in archive mode, make archive
#        sqlplus -s /nolog << EOF > /dev/null 2>&1
#       conn / as sysdba
#      shutdown immediate
#     startup mount
#    alter database archivelog;
#   alter database open;
#  exit
#EOF
#echo_msg "Database changed to archive log mode"
#exit 1
fi
echo_rec_msg "!/usr/bin/ksh"
#start Archive logs
sqlplus -s /nolog << EOF > /dev/null 2>&1
whenever sqlerror exit 1;
conn / as sysdba
set heading off
set pagesize 0
set linesize 10
col sequence# for 999999999
set feedback off
spool ${TMPFILE}
select sequence# from v\$log where status='CURRENT';
spool off
exit
EOF
if [ $? != 0 ]; then
        echo_msg "Error finding start sequence number."
        exit 2
fi
chmod u+x ${TMPFILE}

SEQ=`cat ${TMPFILE}`
echo_msg "Start Sequence#: ${SEQ}"
echo_msg "Start backing up datafiles"
#
#backup datafiles
sqlplus -s /nolog <<EOF > /dev/null 2>&1
whenever sqlerror exit 1;
conn / as sysdba
set echo off
set feedback off
set heading off
set linesize 2048
set pagesize 0
spool ${TMPFILE}
select tablespace_name||':'||file_name||':'||file_id from
dba_data_files order by tablespace_name;
spool off
exit
```

```
EOF
if [ $? != 0 ]; then
        echo_msg "Error querying for datafiles"
        exit 2
fi
echo_msg "Adding datafiles to recovery file"
echo_rec_msg "Datafiles"
echo_msg "Datafiles backed up: "
chmod u+x ${TMPFILE}

cat ${TMPFILE} | awk -F: '{print $1 " " $2 " " $3}' | while read TBN
DF FID
do
sqlplus -s /nolog <<EOF >/dev/null 2>&1
whenever sqlerror exit 1;
conn / as sysdba
alter tablespace ${TBN} begin backup;
--alter system switch logfile;
exit
EOF

cp ${DF} ${BU_LOC_FILE}/${TBN}_${FID}_${YYMMDD}.bk
#touch ${BU_LOC_FILE}/${TBN}_${FID}_${YYMMDD}.bk


if [ $? -ne 0 ]; then
     echo_msg "Failed to cp datafile: ${DF}"
fi


sqlplus -s /nolog <<EOF >/dev/null 2>&1
whenever sqlerror exit 1;
conn / as sysdba
alter tablespace ${TBN} end backup;
exit
EOF
echo_msg "${DF}"
echo_rec_msg "Tablespace: ${TBN}"
echo_rec "${BU_LOC_FILE}/${TBN}_${FID}_${YYMMDD}.bk" "${DF}"
done

echo_msg "Finished Datafiles."

#Done datafiles
#
#
#backing up the control files
```

```
echo_msg "Backing up the control file to '${BU_LOC_FILE}/'"
sqlplus -s /nolog << EOF > /dev/null 2>&1
whenever sqlerror exit 1;
conn / as sysdba
alter database backup controlfile to '${BU_LOC_FILE}/control.bin';
exit
EOF

if [ $? != 0 ]; then
        echo "Error backing up control file."
        exit 2
fi

sqlplus -s /nolog << EOF > /dev/null 2>&1
whenever sqlerror exit 1;
conn / as sysdba
alter database backup controlfile to trace as
'${BU_LOC_FILE}/control.txt';
exit
EOF

if [ $? != 0 ]; then
        echo "Error backing up controlfile to trace"
        exit 2
fi
echo_msg "Finished controlfile backups"

#adding controlfiles to the recovery
echo_msg "Adding to recovery file"
sqlplus -s /nolog << EOF > /dev/null 2>&1
whenever sqlerror exit 1;
conn / as sysdba
set echo off
set feedback off
set heading off
set pagesize 0
set linesize 2048
spool ${TMPFILE}
select name from v\$controlfile;
spool off
exit
EOF
if [ $? != 0 ]; then
        echo "Error querying database for controlfiles"
        exit 2
fi

echo_rec_msg "Control Files"
echo_rec_msg "Trace: ${BU_LOC_FILE}/control.txt"
cat ${TMPFILE} | while read inputline
     do
          echo_rec "${BU_LOC_FILE}/control.bin" "${inputline}"
     done

echo_msg "Finished Controlfiles."
#
#
#
#Archive logs
sqlplus -s /nolog << EOF > /dev/null 2>&1
whenever sqlerror exit 1;
conn / as sysdba
set heading off
set feedback off
set pagesize 0
set linesize 10
col sequence# for 999999999
alter system archive log current;
spool ${TMPFILE}
select sequence# from v\$log where status='CURRENT';
spool off
exit
EOF
if [ $? != 0 ]; then
        echo "Error finding end sequence number."
        exit 2
fi

secseq=`cat ${TMPFILE}`
echo_msg "End Sequence#: ${secseq}"
echo_msg "Start Archive log backups"

sqlplus -s /nolog <<EOF > /dev/null 2>&1
whenever sqlerror exit 1;
conn / as sysdba
set heading off
set pagesize 0
set linesize 2048
set feedback off
spool ${TMPFILE}
select name||':'||sequence# from v\$archived_log where (sequence#
between ${SEQ} and ${secseq}) and
     resetlogs_id = (select resetlogs_id from v\$database_incarnation
where status='CURRENT');
spool off
exit
```

```
EOF
if [ $? != 0 ]; then
        echo "Error querying database for archived logs."
        exit 2
fi

echo_rec_msg "Archive Logs"
cat ${TMPFILE} | awk -F: '{print $1 " " $2}' | while read AL SN
do
     cp ${AL} ${BU_LOC_FILE}/ar_${SN}.arc

     if [ $? -ne 0 ]; then
                echo_msg "Failed to cp archive log : sequence num
${SN}"
        fi


     echo_rec "${BU_LOC_FILE}/ar_${SN}.arc" "${AL}"
done
echo_msg "Finished Archive Logs"
echo_msg "Start network files backups"

#
# Network Files
#
#tnsnames
echo_rec_msg "Network Files"
if [ -f ${ORACLE_HOME}/network/admin/tnsnames.ora ]; then
     echo_msg "Copied tnsnames from default Oracle Home"
     cp ${ORACLE_HOME}/network/admin/tnsnames.ora
${BU_LOC_FILE}/tnsnames.ora

     if [ $? -ne 0 ]; then
                echo_msg "Failed to cp tnsnames.ora"
        fi


     echo_rec "${BU_LOC_FILE}/tnsnames.ora"
"${ORACLE_HOME}/network/admin/tnsnames.ora"
else
     if [ -f /etc/tnsnames.ora ]; then
          echo_msg "Copied tnsnames from /etc"
          cp /etc/tnsnames.ora ${BU_LOC_FILE}/tnsnames.ora

          if [ $? -ne 0 ]; then
               echo_msg "Failed to cp tnsnames.ora"
               fi
```

```
           echo_rec "${BU_LOC_FILE}/tnsnames.ora" "/etc/tnsnames.ora"
      fi
fi
#listener
if [ -f ${ORACLE_HOME}/network/admin/listener.ora ]; then
      echo_msg "Copied listener from default Oracle Home"
      cp ${ORACLE_HOME}/network/admin/listener.ora
${BU_LOC_FILE}/listener.ora

      if [ $? -ne 0 ]; then
                echo_msg "Failed to cp listener.ora"
        fi


      echo_rec "${BU_LOC_FILE}/listener.ora"
"${ORACLE_HOME}/network/admin/listener.ora"


else
      if [ -f /etc/listener.ora ]; then
           echo_msg "Copied listener from /etc"
           cp /etc/tnsnames.ora ${BU_LOC_FILE}/listener.ora
           echo_rec "${BU_LOC_FILE}/listener.ora" "/etc/listener.ora"

           if [ $? -ne 0 ]; then
                      echo_msg "Failed to cp listener.ora"
                fi



      fi
fi
#sqlnet
if [ -f ${ORACLE_HOME}/network/admin/sqlnet.ora ]; then
      echo_msg "Copied sqlnet from default Oracle Home"
      cp ${ORACLE_HOME}/network/admin/sqlnet.ora
${BU_LOC_FILE}/sqlnet.ora

      if [ $? -ne 0 ]; then
           echo_msg "Failed to cp sqlnet.ora"
        fi

      echo_rec "${BU_LOC_FILE}/sqlnet.ora"
"${ORACLE_HOME}/network/admin/sqlnet.ora"
else
      if [ -f /etc/sqlnet.ora ]; then
           echo_msg "Copied sqlnet from /etc"
           cp /etc/tnsnames.ora ${BU_LOC_FILE}/sqlnet.ora
           if [ $? -ne 0 ]; then
                echo_msg "Failed to cp sqlnet.ora"
           fi

           echo_rec "${BU_LOC_FILE}/sqlnet.ora" "/etc/sqlnet.ora"
      fi
fi
echo_msg "Network files done"
echo_msg "Start pfile backup"
#
#Pfile
#
sqlplus -s /nolog << EOF > /dev/null 2>&1
whenever sqlerror exit 1;
conn / as sysdba
create pfile='${BU_LOC_FILE}/init.ora' from spfile;
exit
EOF
if [ $? != 0 ]; then
        echo "Error creating pfile copy"
        exit 2
fi
echo_rec_msg "Pfile"
echo_rec_msg "Pfile location: ${BU_LOC_FILE}/init.ora "
echo_msg "Pfile backup done"
#
#
#
echo_msg "Script Ended"
```

Function Script

```
#!/usr/bin/ksh
echo_msg()
{
      echo "`date +'%Y%m%d_%H:%M:%S'` : $1" | tee -a ${LOGFILE}
}

echo_rec_msg()
{
      echo "#$1" >>  ${REC_FILE}
}

echo_rec()
{
      echo "#cp $1 $2" >> ${REC_FILE}
}
```

Appendix F – Extra Backup Documentation

Backup

Cold Backup

Cold backups are offline backups, where the database is unavailable to the users. It can be achieved while the database is in both archive and no archive log mode.

2. Run the following syntaxes to identify the locations of your listed files (pfile, control files, password files
   datafiles, and the network files):
   ```
   Select name from v$datafile;
   Select member from v$logfile;
   Select name from v$controlfile;
   ```
   More database file information:
   ```
   dba_data_files
   ```
3. to force a logfile switch run:
   ```
   Alter system switch logfile;
   ```
4. shut down normal
5. copy the list of files:
   Password file from default location: \dbhome_1\database
   Pfile: C:\tea\disk1\admin\initTEA12C.ora
   Network files: "listener.ora, sqlnet.ora, tnsnames.ora" from \dbhome_1\network\admin
   Log files: all the RDO log files
   Control files: all the CTL files
6. startup
   Pfile=C:\tea\disk1\admin\initTEA12C.ora;
7. create SP file
   Create spfile=’ C:\ tea\disk5\teaspfile.ora’ from pfile=’ C:\tea\disk1\admin\initTEA12C.ora’;

Flashback

Set flash backup point on, to revert the entire database to a prior point in time. The database must be in archivelog mode and the fast recovery area must be configured. These flashback logs are used to back up the database to specific time.

Example:

```
startup mount;
alter database flashback on;
alter database open;
select flashback_on from v$database;
select current_scn from v$database;

alter pluggable database orclpdb open;
conn bob/bob@orclpdb
create table test_flashback(
  flash# number(1),
  txt varchar2(10)
);
insert into test_flashback values (1,'king');
select * from test_flashback;
commit;
```

```
conn sys/oracle as sysdba
Shutdown;
startup mount;
flashback database to scn 13994206;
alter database open resetlogs; -- change log sequence to 1,
and online redo logs are given a new time stamp and SCN
archive log list; -- will start from sequence 1

alter pluggable database orclpdb open;
conn bob/bob@orclpdb
select * from test_flashback; --table is gone, flashback
succeed
```

Read-Only Backups

1. Alter tablespace query_data read only;
2. Checkpoint is performed for all data files associated with the tablespace
   The file headers are then frozen with the current SCN
3. Must back up all of the data files for the tablespace
4. The DBWO process writes only to data files whose tablespaces are in read-write mode, and normal checkpoints occur on these files

Recovery & Restore

Data Recovery Advisor

Diagnosing and Repairing database failure

1. Use the DRA (Data Recovery Advisor) to list the database failure
2. Obtain an advice how to repair a failure using DRA
3. Repair a failure using DRA

```
C:\> rman target sys@bur as sysdba
RMAN> LIST FAILURE;
RMAN> ADVICE FAILURE;
RMAN> REPAIR FAILURE;

     SQL> show parameter DIAG
     SQL> show parameter dump_dest


     RMAN> validate database;
```

RMAN Recovering in NO-ARCHIVELOG

In no-archive log mode we have to mount the instance

1. Log in to RMAN
2. Shutdown the database
3. Mount the database to create a consistent backup
4. List the backups
5. Recover the database

```
C:\> rman target=sys@bur
RMAN > shutdown immediate;
```

```
          SQL > STARTUP MOUNT pfile= C:\BUR\ Burpfile.ora
          RMAN > list backup;
          RMAN > recover database;
```

Recovering Pfile Loss

1. Connect to sys
2. Log in to RMAN
3. Start up the database. The database will be started with a dummy Pfile
4. Since the database in a dummy mode restore the Spfile the autobackup, Specify the location of the FRA
5. Shut own the database and then restart it with the restored spfile

```
          SQL> conn sys as sysdba
          SQL> host
          C:\> rman target sys@cyan as sysdba
          RMAN> startup;
          RMAN> restore spfile from autobackup recovery area
          'c:……fast recovery area'
          RMAN> shutdown immediate
          RMAN> startup
```

Auxiliary database

1. Connect to the auxiliary database
2. On separate command prompt connect to the target database

```
          C:\> rman auxiliary sys/Elcaro1! @aux
          C:\> rman target sys@bur as sysdba
```

We can override where RMAN will write the auto backup control file and its name using the configure command. For example:

```
          RMAN> configure controlfile autobackup format for device
     type disk to '/u01/app/oracle/backup/autobackup/controlfile_%F';
```

to set the directory and file format to the default

```
          RMAN> configure controlfile autobackup format for device
     type disk clear;
```

RMAN Configuration

CONFIGURE DEVICE TYPE 'SBT_TAPE' PARALLELISM 3 BACKUP TYPE TO BACKUPSET;

Specify channels for device type

```
          RMAN> configure channel 1 device type disk format
          '/disk1/%U';
          RMAN> configure channel 2 device type disk format
          '/disk2/%U';
          RMAN> configure channel 3 device type disk format
          '/disk3/%U';
```

To how the errors and the status of data files

```
          SQL> select file#, status, error, recover from
     v$datafile_header;
```

The V$RECOVER_FILE view displays the status of files needing media recovery

```
          SQL> select file#, online_status, error from
     v$recover_file;
```

Use restore…preview

```
          RMAN> restore database preview;
          RMAN> restore database from tag TAG20120722T203338 preview;
          RMAN> restore datafile 1, 2, 3, 4 preview;
          RMAN> restore archivelog all preview;
          RMAN> restore archivelog from time 'sysdate - 1' preview;
          RMAN> restore archivelog from scn 850000 preview;
          RMAN> restore archivelog from sequence 29 preview;
```

If you use a media manager that supports vaulted backups, you can use preview recall to recall media from remote storage

```
          RMAN> restore database preview recall;
```

Using restore . . . validate

The validate clause works with any restore command. For example, this validates the backup files required to restore the database:

```
          RMAN> restore database validate;
```

Incomplete Recovery

Missing a control file

1. Set the database in no mount
2. Restore the missing or the corrupted CF
3. Set the database to the mount stage
4. Open the database using resetlogs

```
          SQL> SHUTDOWN IMMEDIATE
          SQL> STARTUP NOMOUNT Pfile= C:\bur\ rcatpfile.ora
          RMAN> Restore controlfile from autobackup;
          RMAN> alter database mount
          RMAN> alter database open resetlogs;
```

Missing Data File

We still can use the DRA to detect the failure

```
          RMAN> list failure
          RMAN> advise failure
```

We can fix the failure by using

```
          RMAN> repair failure
```

Or fix it manually:

5. Set the database in no mount

```
          SQL> SHUTDOWN IMMEDIATE
          SQL> STARTUP NOMOUNT Pfile= C:\bur\ rcatpfile.ora
```

6. Restore the missing datafile (assuming that the missing one is datafile 2)

```
      RMAN> Restore datafile 2 from auto backup
```

   OR we can restore it from a specific log sequence number

   a. Run the following statement to view the sequence number
   b. Restore the entire database from the backup before the missing archive log
   c. Recover the database through the last available logfile

```
      SELECT sequence#,
            first_change#,
            first_time,
            open_mode
      FROM v$databse;
      RMAN> RSETORE DATABASE UBTILL SEQUENCE 60
      RMAN> RECOVER DATABASE UBTILL SEQUENCE 60
      RMAN> recover datafile 2;
```

7. Verify by checking oldest and next log sequence
8. Open the database using resetlogs

```
      SQL> archive log list
      RMAN> alter database open resetlogs;
```

Crosscheck Archive logs using Recovery Catalog

Verifies the status of backups and copies recorded in the RMAN repository against media such as disk or tape

1. Connect to catalog
2. Run the statement

```
C:\> rman target sys@bur catalog rcat_owner@rcat
RMAN> CROSSCHECK ARCHIVELOG ALL;
```

Recover to Image Copies

```
RMAN> recover copy of datafile {n|'file_name'}
```

--if run these commands daily

recover copy of database with tag 'daily_inc';

backup incremental level 1 for recover of copy with tag 'daily_inc' database;

Perform a fast switch to image copies

1. Take data files offline
2. Use the SWITCH TO ...COPY command to switch to image copies
3. Recover data files
4. Bring data files online

--now the data files are recovered and usable in their new location

--optionally, do the following to put the files back into their original location

5. Create an image copy of the data file in the original location
6. Take data files offline
7. SWITCH TO ...copy
8. Recover data files
9. Bring data files online

```
      SQL> switch datafile '...' to copy;
```

--use SET NEWNAME for switching files

--the set newname command can be used only inside a RUN block

```
RMAN> run{
     allocate channel dev1 device type disk;
     allocate channel dev2 device type sbt;
     SQL "alter tablespace users offline immediate";
     set newname for datafile '...' to '...';
     restore tablespace users;
     switch datafile all;
     recover tablespace users;
     SQL "alter tablespace users online";}
```

RC_VIEWS

Some examples of rc_views:

- RC_ARCHIVED_LOG
- RC_BACKUP_ARCHIVELOG_DETAILS
- RC_BACKUP_CONTROLFILE
- RC_BACKUP_DATAFILE
- RC_BACKUP_REDOLOG
- RC_BACKUP_CORRUPTION
- RC_DATABASE
- RC_CONTROLFILE_COPY
- RC_DATAFILE
- RC_DATAFILE_COPY
- RC_LOG_HISTORY
- RC_TABLESPACES

And more…

```
SQL> select object_name from user_objects where object_name like
'RC_%';
```

Restore point

- must have SYSDBA privileges
- must have flash recovery area
- must be in ARCHIVE LOG MODE
- flashback should be on

```
     SQL> create restore point [name_of_restore_point]
[guarantee] flashback database;
```

- to view restore points:

```
     SQL> select * from v$restore_point;
```

To restore

```
SQL> select current_scn from v$database;
SQL> shutdown immediate;
SQL> startup mount;
SQL> flashback database to restore point [name_of_restore_point];
SQL> alter database open resetlogs;
SQL> select current_scn from v$database;        --compare SCN# to
the SCN from v$restore_point
```

Error Handling

Tips

- system and undo cannot be offline when database is running
  - if using GUI, sysaux cannot be offline either
- to find files that need recovery:

```
select name, error from v$datafile join v$recover_file using (file#);
```

- useful commands to monitor block change tracking

```
select filename,status,bytes from v$block_change_tracking;
select file#,
        avg(datafile_blocks),
        avg(blocks_read),
        avg(blocks_read/datafile_blocks)*100 as PCT_READ_FOR_BACKUP,
        avg(blocks)
from v$backup_datafile where used_change_tracking='YES' and incremental_level>0
group by file#;
```

- Report on backups

```
RMAN>
        list
        report
        report need backup
        report obsolete
```

  SQL dynamic views

```
v$backup_set
v$backup_piece
v$datafile_copy
v$backup_files
```

  SQL Data recovery advisor Views

```
v$ir_failure
v$ir_manual_checklist
v$ir_repair
v$ir_failure_set
```

- delete expired: remove only files whose status in the repository is "expired"
- delete obsolete: delete backups that are no longer needed
- compressing backups – may save space, however will increase MTTR
- debug option

```
RMAN> target / catalog rman/rman debug trace trace.log
RMAN> run {
        debug on;
        allocate channel c1 type disk;
        backup datafile 3;
        debug off;
        backup datafile 4;}
```

Common Errors

File Loss can be caused by:

- User error
- Application error
- Media failure

A non-critical file can be addressed by:

- Creating a new file
- Rebuild the file/having scripts (ex. indx)
- Recover the lost or damaged file
- Import from backups

Data repair techniques:

Physical Failure (missing or corrupted data file):

- Data recovery advisor
- Data file media recovery
- Block recovery

Logical Failure (Application or user error)

- Logical flashback features
- Oracle flashback database
- Point-in-time recovery (in RMAN)
    - DBPITR, PITR, TPITR

Control File Loss

- If control files are stored in ASM disk group, recovery options:
    - Using Enterprise Manager:
        - Startup nomount and use RMAN
          Restore control file from '…';
- If control files are stored as regular file system files;
    - Shutdown
    - Replace lost control file from existing control file copies
- RMAN from Auto backup

```
startup nomount; --without recovery catalog: rman> set dbid <nn>
restore controlfile from autobackup;
alter database mount;
recover database; --go back in time
alter database open resetlogs; --new db incarnation
--restore the spfile and the control file
--with recovery catalog: db in nomount state
RMAN> restore controlfile;
RMAN> restore controlfile ...to <dest>
--loss of spfile and controlfile:
set the dbid or use recovery catalog
restore the spfile from the autobackup
start the instance with the restored spfile
RMAN > restore controlfile from autobackup;
mount the db with the restored control file
restore and recover the db
```

RMAN > alter database open resetlogs;

Loss of Redo Log File

1. Determine whether there is a missing log file by examining the alert log
2. Alter database drop logfile member '…';
   Alter database add logfile member '…' to group <#>;
   OR
   Replace the missing control file by creating a copy from existing other members
3. Rename the missing file if the media failure is due to the loss of a disk drive or controller
4. If the group has already been archived, or if you are in no-archive log mode, you may choose to solve the problem by clearing the log group to recreate the missing file or files
   Alter database clear logfile group <#>; -- if no members exist

Loss of Data file in No-Archive log mode

1. Shutdown
2. Restore from redo logs
3. Startup
4. Re-enter all changes since last backup

- Might be able to import using datapump if it exists

Loss of a system-critical data file in Archive log mode

1. Shutdown abort
2. Startup mount
3. Restore and recover the missing data file
4. Alter database open

Loss of No-logging database objects

Objects created with NOLOGGING cannot be recovered (even if the database is in archive log mode)

```
SQL> create table abc_copy nologging;
SQL> insert /*+ append */ into abc_copy select * from abc;
--no redo is generated for this particular insert statement
--cannot recover this transaction on abc table
```

Loss of PWDFILE

Recreate a password authentication file using orapwd

1. orapwd file=?/database/filename password=password entries=max_users
   other options: ASM/FORMAT/SYSBACKUP/SYSDG/SYSKM/DELETE/FORCE
2. add users to the password file and assign appropriate privileges to each user
   desc v$pwfile_users;

Copy over password file from backup to the correct location in oracle home.

Disaster recovery

1. Restore an autobackup of the server parameter file
2. Startup nomount;
3. Restore the control file from autobackup
4. Alter database mount;
5. Restore the data files
6. Recover the data files

7. Alter database open resetlogs;

Consider on instance recovery in the plan

Increase the MTBF, the mean time between failures. Increase the average time between a failure and the next expected one that may occur

Decrease the MTTR, the mean time to recover. Decrease the time it takes to run a repair after a data loss.

We can add a MTTR parameter to the pfile to specify the seconds to perform an instance recovery

```
show parameter fast_start_mttr_target
show parameter log_checkpoint_timeout
show parameter log_checkpoint_interval
SELECT TARGET_MTTR,
        ESTIMATED_MTTR,
        CKPT_BLOCK_WRITES
FROM V$INSTANCE_RECOVERY;
```

RMAN Backup in NO-ARCHIVELOG

1. Shutdown the database
2. Mount the database to create a consistent backup
3. Backup the database
4. Open the database
5. Verify the created backup
6. Ensure that we're saving spaces by removing duplicates in case they are existed

```
RMAN> shutdown immediate;
RMAN> startup mount;
RMAN> backup database;
RMAN> ALTER DATBSE OPEN;
RMAN> list backup;
RMAN> delete obsolete;
RMAN> exit
```

User-managed backup in No-archivelog Mode

Multiplex CF by creating multiple 16 redo log groups with 2 members for each group in different location, 2 copies of archive logs in 2 separate disks. One of the locations should be local

Virtual Private Catalog (VPC)

Create the owner of the VPC

1. Use sqlplus, connect to the recovery catalog database
2. Create the VPC owner
3. Grant recovery_catalog_owner role to the user
4. Using RMAN, connect to the recovery catalog as the base recovery catalog owner
5. Grant privileges (access to the metadata/ability to register new target database) to the VPC owner

Create the VPC

1. Using RMAN, connect to the recovery catalog db as the VPC owner
2. Create the VPC by "create virtual catalog" command

```
--revoke access to the metadata
RMAN> revoke catalog for database netclass from vpc1;
--to drop a virtual private catalog
RMAN> drop catalog;
```

Register a database with a VPC and store backup metadata

1. Using RMAN, connect to the database that you want to register as target
2. Register the database with the VPC owner by using the "register database" command
3. Use the "backup" command to back up the database and store metadata related to the backup in the VPC

Using Nologging

While reloading use keyword Nologging and parallel to make the reloading faster.
Nologging avoids generating redo (used for tuning)

Appendix G – Reports from TEA Database

Workshop 3

```
SQL> conn sys/Elcaro1! as sysdba
Connected to an idle instance.
SQL> show user
USER is "SYS"
SQL> SET heading OFF
SQL> SELECT 'Instance name: ', instance_name FROM v$instance;

Instance name:  tea

SQL>
SQL> startup nomount pfile='C:\TEA\DISK1\ADMIN\initTEA12c.ora'
ORACLE instance started.

Total System Global Area 5167382528 bytes
Fixed Size                  8757568 bytes
Variable Size             956305088 bytes
Database Buffers         4194304000 bytes
Redo Buffers                8015872 bytes

SQL> spool off
--------------------------------------Creating database---------------
-----
SQL> SET feedback ON
SQL> --Script for DB creation
SQL>
SQL> show user
USER is "SYS"
SQL> SET HEADING OFF
SQL>
SQL> SELECT 'Instance name: ',instance_name
  2  FROM v$instance;

Instance name:  tea

1 row selected.

SQL>
SQL> CREATE DATABASE TEA
  2       USER SYS IDENTIFIED BY oracle
  3       USER SYSTEM IDENTIFIED BY oracle
  4       LOGFILE GROUP 1 ('C:\TEA\DISK1\redo01a.log',
'C:\TEA\Disk2\redo01b.log', 'C:\TEA\Disk3\redo01c.log') SIZE 10M,
  5                  GROUP 2 ('C:\TEA\DISK1\redo02a.log',
'C:\TEA\Disk2\redo02b.log', 'C:\TEA\Disk3\redo02c.log') SIZE 10M,
  6                  GROUP 3 ('C:\TEA\DISK1\redo03a.log',
'C:\TEA\Disk2\redo03b.log', 'C:\TEA\Disk3\redo03c.log') SIZE 10M
  7       MAXINSTANCES 1
  8       MAXLOGHISTORY 1
  9       MAXLOGFILES 16
 10       MAXLOGMEMBERS 3
 11       MAXDATAFILES 1024
 12       CHARACTER SET AL32UTF8
 13  NATIONAL CHARACTER SET AL16UTF16
 14  DATAFILE 'C:\TEA\DISK1\system01.dbf'
 15       SIZE 500M  AUTOEXTEND ON NEXT 15M MAXSIZE 1000M
 16  SYSAUX DATAFILE 'C:\TEA\Disk4\sysaux01.dbf'
 17    SIZE 400M AUTOEXTEND ON NEXT 10M MAXSIZE 800M
 18     DEFAULT TEMPORARY TABLESPACE temp01
 19     TEMPFILE 'C:\TEA\Disk2\temp01.dbf'
 20     SIZE 200M AUTOEXTEND ON NEXT 5M MAXSIZE 800M
 21   UNDO TABLESPACE undotbs01
 22     DATAFILE 'C:\TEA\DISK3\undo01.dbf'
 23     SIZE 200M AUTOEXTEND ON NEXT 10M MAXSIZE 400M
 24       USER_DATA TABLESPACE userdata01
 25            DATAFILE 'C:\TEA\DISK3\user_data01.dbf'
 26            SIZE 400M AUTOEXTEND ON NEXT 30M MAXSIZE 800M;

Database created.

SQL>
SQL> CREATE TABLESPACE userdata02
  2       DATAFILE 'C:\TEA\DISK4\user_data02.dbf'
  3       SIZE 400M AUTOEXTEND ON NEXT 30M MAXSIZE 800M;

Tablespace created.

SQL> CREATE UNDO TABLESPACE undotbs02
  2       DATAFILE 'C:\TEA\DISK4\undo02.dbf'
  3       SIZE 200M AUTOEXTEND ON NEXT 10M MAXSIZE 400M;

Tablespace created.

SQL> CREATE TABLESPACE INDX
  2       DATAFILE 'C:\TEA\DISK2\indx.dbf'
  3       SIZE 300M AUTOEXTEND ON NEXT 20M MAXSIZE 800M;

Tablespace created.

SQL> CREATE TABLESPACE ADHOC
  2       DATAFILE 'C:\TEA\DISK3\adhoc.dbf'
  3       SIZE 50M AUTOEXTEND ON NEXT 10M MAXSIZE 100M;

Tablespace created.

SQL> CREATE TABLESPACE RO
  2       DATAFILE 'C:\TEA\DISK1\ro.dbf'
  3       SIZE 50M AUTOEXTEND ON NEXT 10M MAXSIZE 100M;

Tablespace created.
```

```
SQL> alter tablespace system
  2  add datafile 'c:\tea\disk2\system02.dbf'
  3  size 500m autoextend on next 15m maxsize 800m;

Tablespace altered.

SQL>
SQL> SET echo off
-------------------- Before catalogue & catproc--------------------
SQL> SET Linesize 100
SQL> SET pagesize 50
SQL> SET echo on
SQL>
SQL> col comp_id for a20
SQL> col comp_name for a30 WORD_WRAPPED
SQL> col status for a10
SQL> col version for a12
SQL> sho user
USER is "SYS"
SQL>
SQL> SELECT comp_name, comp_id, status, version
  2  FROM dba_registry
  3  ORDER BY 1;
FROM dba_registry
     *
ERROR at line 2:
ORA-00942: table or view does not exist

SQL> set termout off
SQL> set echo off
SQL> spool off
--------------------------After catalogue & catproc-------------------
-
SQL>
SQL> col comp_id for a20
SQL> col comp_name for a30 WORD_WRAPPED
SQL> col status for a10
SQL> col version for a12
SQL>
SQL> SET echo ON
SQL> SET termout ON
SQL>
SQL> SELECT comp_name, comp_id, status, version
  2  FROM dba_registry
  3  ORDER BY 1;

COMP_NAME                      COMP_ID              STATUS     VERSION
------------------------------ -------------------- ---------- -------
-----
Oracle Database Catalog Views  CATALOG              VALID
12.2.0.1.0
```

```
Oracle Database Packages and   CATPROC              VALID
12.2.0.1.0
Types

Oracle XML Database            XDB                  VALID
12.2.0.1.0

3 rows selected.

SQL> show user
USER is "SYS"
SQL> spool off
--------------------------Before pupbld--------------------
SQL> conn system/oracle
Connected.
SQL> desc PRODUCT_PRIVS;
ERROR:
ORA-04043: object PRODUCT_PRIVS does not exist


SQL> sho user
USER is "SYSTEM"
SQL> set termout off
SQL> set echo off
SQL> set echo on
SQL> set echo off
SQL>
SQL>
@C:\app\Administrator\product\12.2.0\dbhome_1\sqlplus\admin\pupbld.sql

-----------------------------------After pupbld--------------------
SQL> set echo on
SQL> set termout on
SQL> desc product_privs;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------
----------
 PRODUCT                                   NOT NULL VARCHAR2(30)
 USERID                                             VARCHAR2(128)
 ATTRIBUTE                                          VARCHAR2(240)
 SCOPE                                              VARCHAR2(240)
 NUMERIC_VALUE                                      NUMBER(15,2)
 CHAR_VALUE                                         VARCHAR2(240)
 DATE_VALUE                                         DATE
 LONG_VALUE                                         LONG

SQL>
SQL> SET HEADING ON
SQL> TTITLE OFF
SQL> BTITLE OFF
SQL> SET linesize 100
SQL> SET pagesize 100
SQL> spool off
----------------------------Verifying Database--------------------
SQL> @c:\cyan\v
SQL> SET Linesize 40
SQL> SET pagesize 15
SQL> set echo off
USER is "SYSTEM"

Tue Nov 26                     page    1
                 Verify

Database name:  TEA
Instance name:  tea

Date:  26-NOV-19
USER is "SYSTEM"

Tue Nov 26                                                  page    1
                                Files

STATUS  NAME
------- --------------------------------------------------
        C:\TEA\DISK1\REDO01A.LOG
        C:\TEA\DISK1\REDO02A.LOG
        C:\TEA\DISK1\REDO03A.LOG
        C:\TEA\DISK2\REDO01B.LOG
        C:\TEA\DISK2\REDO02B.LOG
        C:\TEA\DISK2\REDO03B.LOG
        C:\TEA\DISK3\REDO01C.LOG
        C:\TEA\DISK3\REDO02C.LOG
        C:\TEA\DISK3\REDO03C.LOG
SYSTEM  C:\TEA\DISK1\SYSTEM01.DBF
SYSTEM  C:\TEA\DISK2\SYSTEM02.DBF
ONLINE  C:\TEA\DISK1\RO.DBF
ONLINE  C:\TEA\DISK2\INDX.DBF
ONLINE  C:\TEA\DISK3\ADHOC.DBF
ONLINE  C:\TEA\DISK3\UNDO01.DBF
ONLINE  C:\TEA\DISK3\USER_DATA01.DBF
ONLINE  C:\TEA\DISK4\SYSAUX01.DBF
ONLINE  C:\TEA\DISK4\UNDO02.DBF
ONLINE  C:\TEA\DISK4\USER_DATA02.DBF
Control C:\TEA\DISK1\CONTROL01.CTL
Control C:\TEA\DISK2\CONTROL02.CTL
Control C:\TEA\DISK3\CONTROL03.CTL
Control C:\TEA\DISK5\CONTROL04.CTL

Tue Nov 26                                                  page    1
                              Log Files

NAME       VALUE
---------- --------------------------------------------------
Active     0
```

```
Problem
Count

Active     0
Incident
Count

ADR Base   C:\TEA\DISK5
ADR Home   C:\TEA\DISK5\diag\rdbms\tea\tea
Diag Alert C:\TEA\DISK5\diag\rdbms\tea\tea\alert
Diag Cdump C:\TEA\DISK5\diag\rdbms\tea\tea\cdump
Health     C:\TEA\DISK5\diag\rdbms\tea\tea\hm
Monitor

Diag       C:\TEA\DISK5\diag\rdbms\tea\tea\incident
Incident

Diag Trace C:\TEA\DISK5\diag\rdbms\tea\tea\trace
Default    C:\TEA\DISK5\diag\rdbms\tea\tea\trace\tea_ora_2224
Trace File .trc

Diag       TRUE
Enabled


Tue Nov 26                                                page    1
                              Log Files

    GROUP# STATUS              MEMBERS
---------- ---------------- ----------
         1 INACTIVE                  3
         2 INACTIVE                  3
         3 CURRENT                   3

Tue Nov 26                                                page    1
                             TableSpaces

 TS# Loc                                                  TS Name
---- ---------------------------------------------------- ----------
   0 C:\TEA\DISK1\SYSTEM01.DBF                            SYSTEM
     C:\TEA\DISK2\SYSTEM02.DBF                            SYSTEM
   1 C:\TEA\DISK4\SYSAUX01.DBF                            SYSAUX
   2 C:\TEA\DISK3\UNDO01.DBF                              UNDOTBS01
   4 C:\TEA\DISK3\USER_DATA01.DBF                         USERDATA01
   5 C:\TEA\DISK4\USER_DATA02.DBF                         USERDATA02
   6 C:\TEA\DISK2\INDX.DBF                                INDX
   7 C:\TEA\DISK3\ADHOC.DBF                               ADHOC
   8 C:\TEA\DISK1\RO.DBF                                  RO
   9 C:\TEA\DISK4\UNDO02.DBF                              UNDOTBS02

Tue Nov 26                                                page    1
                             TableSpaces
```

```
TS Name              STATUS    LOGGING   RETENTION   Block Size
-------------------- --------- --------- ----------- ----------
ADHOC                ONLINE    LOGGING   NOT APPLY         8192
INDX                 ONLINE    LOGGING   NOT APPLY         8192
RO                   ONLINE    LOGGING   NOT APPLY         8192
SYSAUX               ONLINE    LOGGING   NOT APPLY         8192
SYSTEM               ONLINE    LOGGING   NOT APPLY         8192
TEMP01               ONLINE    NOLOGGING NOT APPLY         8192
UNDOTBS01            ONLINE    LOGGING   NOGUARANTEE       8192
UNDOTBS02            ONLINE    LOGGING   NOGUARANTEE       8192
USERDATA01           ONLINE    LOGGING   NOT APPLY         8192
USERDATA02           ONLINE    LOGGING   NOT APPLY         8192

Tue Nov 26                     page    1
                  SGA

NAME                                VALUE
------------------------- --------------
Fixed Size                8757568
Variable Size             956305088
Database Buffers          4194304000
Redo Buffers              8015872

Tue Nov 26                                                   page    1
                              Parameters

NAME                           VALUE
------------------------------ -----------------------------------
compatible                     12.2.0
db_block_size                  8192
db_name                        TEA
db_recovery_file_dest          C:\TEA\DISK5
db_recovery_file_dest_size     10009706496
diagnostic_dest                C:\TEA\DISK5
instance_name                  TEA
processes                      150
sga_target                     5167382528
undo_management                AUTO
undo_retention                 1800
undo_tablespace                UNDOTBS01
SQL> TTITLE OFF
SQL> BTITLE OFF
SQL> SET linesize 60
SQL> SET pagesize 60
SQL> SET long 80
SQL> spool off
```

Workshop 4

```
SQL> @C:\TEA\DISK1\ADMIN\REPORTS.SQL
SQL> --Reports
SQL> DESC DONOTTOUCH.AFFILIATION
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 AFFILIATIONCODE                NOT NULL VARCHAR2(10)
 AFFILIATIONDESC                         VARCHAR2(100)

SQL> DESC DONOTTOUCH.CLASS
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 CLASSCODE                      NOT NULL VARCHAR2(5)
 CLASSDESC                      NOT NULL VARCHAR2(100)

SQL> DESC DONOTTOUCH.COMMISSION
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 LISTNO                         NOT NULL NUMBER(38)
 COMMISSIONNO                            NUMBER(2)
 COMMISSIONAMOUNT               NOT NULL NUMBER(10,2)
 EMPID                          NOT NULL NUMBER(38)

SQL> DESC DONOTTOUCH.COMMISSIONTYPE
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 COMMISSIONNO                   NOT NULL NUMBER(2)
 COMMISSIONTYPE                 NOT NULL VARCHAR2(25)

SQL> DESC DONOTTOUCH.CREDITCARD
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 CARDTYPENO                     NOT NULL NUMBER(2)
 CREDITDESC                     NOT NULL VARCHAR2(50)

SQL> DESC DONOTTOUCH.CUSTOMER
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 CUSTOMERID                     NOT NULL NUMBER(38)
 EMPID                                   NUMBER(38)
 FIRSTNAME                      NOT NULL VARCHAR2(15)
 LASTNAME                       NOT NULL VARCHAR2(20)
 HPHONE                                  VARCHAR2(15)
 BPHONE                                  VARCHAR2(15)
 ADDRESS                        NOT NULL VARCHAR2(100)
 CITYCODE                                NUMBER(38)
 ZIP_POSTAL                     NOT NULL VARCHAR2(10)

SQL> DESC DONOTTOUCH.CUSTREWARDCARD
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 CARDNO                         NOT NULL VARCHAR2(20)
 CARDTYPE                       NOT NULL NUMBER(3)
 CUSTOMERID                              NUMBER(38)
```

```
SQL> DESC DONOTTOUCH.CUSTCREDIT
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 CUSTOMERID                     NOT NULL NUMBER(38)
 CREDITCARDNO                   NOT NULL VARCHAR2(20)
 EXPDATE                                 DATE
 CARDTYPENO                              NUMBER(2)

SQL> DESC DONOTTOUCH.DESTINATION
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 DESTINATIONCODE                NOT NULL VARCHAR2(5)
 DESTINDESC                     NOT NULL VARCHAR2(100)

SQL> DESC DONOTTOUCH.DESCRIPTION
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 DESCRIPTIONNO                  NOT NULL NUMBER(38)
 DESCRIPTION                    NOT NULL VARCHAR2(100)

SQL> DESC DONOTTOUCH.EMPLOYEE
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 EMPID                          NOT NULL NUMBER(38)
 FIRSTNAME                      NOT NULL VARCHAR2(15)
 LASTNAME                       NOT NULL VARCHAR2(20)
 HIREDATE                       NOT NULL DATE
 PHONE                                   VARCHAR2(15)
 EMAIL                                   VARCHAR2(50)

SQL> DESC DONOTTOUCH.FEETYPE
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 FEECODE                        NOT NULL CHAR(3)
 FEETYPEDESC                    NOT NULL VARCHAR2(50)
 FEEAMOUNT                               NUMBER(5,2)

SQL> DESC DONOTTOUCH.INVOICE
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 INVOICENO                      NOT NULL NUMBER(38)
 ITINERARYNUM                   NOT NULL NUMBER(38)
 PAYMENTTYPE                             NUMBER(2)
 BILLINGDATE                    NOT NULL DATE
 BILLAMOUNT                              NUMBER(12,2)
 REMAINDER                               NUMBER(12,2)
 SALEDATE                       NOT NULL DATE
 FEECODE                                 CHAR(3)
 CREDITCARD                              VARCHAR2(20)
 CUSTID                                  NUMBER

SQL> DESC DONOTTOUCH.LOCATION
```

```
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 CITYCODE                       NOT NULL NUMBER(38)
 CITYNAME                       NOT NULL VARCHAR2(50)
 STATEPROV                               VARCHAR2(20)
 COUNTRYNAME                    NOT NULL VARCHAR2(20)

SQL> DESC DONOTTOUCH.OPTIONALCUSTDETAIL
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 CUSTOMERID                     NOT NULL NUMBER(38)
 BIRTHDATE                               DATE
 WEDDING                                 DATE
 EMAIL                                   VARCHAR2(50)
 WHEELCHAIR                              CHAR(1)

SQL> DESC DONOTTOUCH.PAYMENTTYPE
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 PAYMENTTYPE                    NOT NULL NUMBER(2)
 PAYMENTDESC                    NOT NULL VARCHAR2(50)

SQL> DESC DONOTTOUCH.PRODUCT
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 PRODUCTCATNO                   NOT NULL NUMBER(3)
 SUPPLIERNO                     NOT NULL NUMBER(38)
 COMPANY                        NOT NULL VARCHAR2(100)
 TAXTYPE                                 NUMBER(3)
 TAXRATE                                 NUMBER(3,2)
 PRICE                                   NUMBER

SQL> DESC DONOTTOUCH.PRODUCTCAT
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 PRODUCTCATNO                   NOT NULL NUMBER(3)
 PRODUCTCATEGORY                         VARCHAR2(100)

SQL> DESC DONOTTOUCH.REWARDCARD
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 CARDTYPE                       NOT NULL NUMBER(3)
 REWARDCARDDESC                 NOT NULL VARCHAR2(50)

SQL> DESC DONOTTOUCH.SUPPLIER
 Name                           Null?    Type
 ------------------------------ -------- --------------------
 SUPPLIERNO                     NOT NULL NUMBER(38)
 COMPANYNAME                    NOT NULL VARCHAR2(100)
 AFFILIATIONCODE                         VARCHAR2(10)
 REP_NO                                  NUMBER(38)
 REPRESENTS                              VARCHAR2(100)
 OFFICENO                               NUMBER(3)
 CONTACTNAME                            VARCHAR2(50)
 ADDRESS                                VARCHAR2(100)
 EMAIL                                  VARCHAR2(50)
 PHONE                                  VARCHAR2(15)
 FAX                                    VARCHAR2(15)
 WEB                                    VARCHAR2(50)
 ZIP_POSTAL                             VARCHAR2(10)
 CITYCODE                               NUMBER(38)

SQL> DESC DONOTTOUCH.TAXTYPE
 Name                          Null?    Type
 ----------------------------- -------- --------------------
 TAXTYPE                       NOT NULL NUMBER(3)
 TAXDESC                       NOT NULL VARCHAR2(100)

SQL> DESC DONOTTOUCH.TRIPPRODUCT
 Name                          Null?    Type
 ----------------------------- -------- --------------------
 LISTNO                        NOT NULL NUMBER(38)
 ITINERARYNUM                           NUMBER(38)
 SUPPLIERNO                             NUMBER(38)
 COMPANY                                VARCHAR2(100)
 PRODUCTCATNO                           NUMBER(3)
 STARTDATE                     NOT NULL DATE
 ENDDATE                       NOT NULL DATE
 BOOKINGNUM                    NOT NULL VARCHAR2(15)
 DESTINATIONCODE                        VARCHAR2(5)
 CLASSCODE                              VARCHAR2(5)
 DESCRIPTIONNO                          NUMBER(38)
 TRIPDESTINATION                        VARCHAR2(50)

SQL> DESC DONOTTOUCH.TRIPTYPE
 Name                          Null?    Type
 ----------------------------- -------- --------------------
 TRIPTYPECODE                  NOT NULL CHAR(2)
 TRIPDESC                      NOT NULL VARCHAR2(50)

SQL> DESC DONOTTOUCH.TRIP
 Name                          Null?    Type
 ----------------------------- -------- --------------------
 ITINERARYNUM                  NOT NULL NUMBER(38)
 TRIPTYPECODE                           CHAR(2)
 CUSTOMERID                             NUMBER(38)
 STARTDATE                              DATE
 ENDDATE                                DATE
 TRAVELLERS                             NUMBER(3)

SQL> DESC DONOTTOUCH.TRIPTRAVELLER
 Name                          Null?    Type
 ----------------------------- -------- --------------------
 ITINERARYNUM                  NOT NULL NUMBER(38)
 TRAVELLERNAME                  NOT NULL VARCHAR2(50)

SQL> SET LINESIZE 50
SQL>
SQL> ------------------------------------------------------
SQL> SET ECHO OFF

Row Counts
----------
       550

1 row selected.


Mon Jan 20                               page    1
                Product Categories

Category
Range      CATEGORY
---------- -----------------------------------
100 - 149  Airline Consolidators
150 - 199  Airlines
200 - 249  Attractions
250 - 299  Car Rentals
300 - 349  Cruise Lines

5 rows selected.


Mon Jan 20                               page    1
                   Affiliation

Row Counts
----------
         7

1 row selected.


Mon Jan 20                               page    1
                   Affiliation

Code     Description
-------- --------------------
ACTA     Association of
         Canadian Travel
         Agents

ACTANEW
ACTANEWP
ACTAPGY
NEW
```

```
5 rows selected.


Mon Jan 20                               page    1
                       Fees

Row Counts
----------
         7

1 row selected.


Mon Jan 20                               page    1
                       Fees

Code Description          Amount
---- -------------------- ------
NC   No Charge                $0
CH   Change                  $15
RF   Refund                  $25
NSF  Insufficient Funds      $25
BK   Booking Charge          $25

5 rows selected.


Mon Jan 20                               page    1
                 Commission Types

Row Counts
----------
         6

1 row selected.


Mon Jan 20                               page    1
                 Commission Types

 No. Type
---- ----------
   1 Booking
   2 Paying
   3 Overdue
   4 Invalid
   5 Cancelled

5 rows selected.
```

```
Mon Jan 20                               page    1
                    Employees

Row Counts
----------
         9

1 row selected.


Mon Jan 20                               page    1
                    Employees

  ID Name
---- ------------------------------
   1 Delton, Janet
   2 Lisle, Judy
   3 Reynolds, Dennis C.
   4 Coville, John
   5 Dahl, Janice W.

5 rows selected.


Mon Jan 20                               page    1
                Destination Codes

Row Counts
----------
        10

1 row selected.


Mon Jan 20                               page    1
                Destination Codes

Code  Description
----- ------------------------------
AFR   Africa
ANZ   Australia and New Zealand
ASIA  Asia
EU    Europe and United Kingdom
MEAST Middle East

5 rows selected.


Mon Jan 20                               page    1
                (Trip)Class Types

Row Counts
```

```
----------
         8

1 row selected.


Mon Jan 20                               page    1
                (Trip)Class Types

Code  Description
----- ------------------------------
BSN   Business Class
DBL   Double
DLX   Delux
ECN   Economy Class
FST   First Class

5 rows selected.


Mon Jan 20                               page    1
                    Trip Types

Row Counts
----------
         3

1 row selected.


Mon Jan 20                               page    1
                    Trip Types

Code  Description
----- ------------------------------
B     Business
G     Group
L     Leisure

3 rows selected.


Mon Jan 20                               page    1
                    COMMISSION

Row Counts
----------
       286

1 row selected.
```

```
Mon Jan 20                               page    1
                    COMMISSION

                 Commission
    LISTNO           Amount Type
---------- ---------------- ----------
         1          $210.00 Unknown
         1          $210.00 Unknown
         2           $24.50 Unknown
         3           $24.50 Unknown
         4           $24.50 Unknown

5 rows selected.


Mon Jan 20                               page    1
                    CREDITCARD

Row Counts
----------
         4

1 row selected.


Mon Jan 20                               page    1
                    CREDITCARD

 No. Card
---- --------------------
   1 AMEX
   2 VISA
   3 MC
   4 DINERS

4 rows selected.


Mon Jan 20
page    1
                                      CUSTOMER

Row Counts
----------
       270

1 row selected.


Mon Jan 20
page    1
                                      CUSTOMER
```

```
      Emp.       Cust
        ID         ID Name            Home        Business    ADDRESS
---------- ---------- --------------- ----------- ----------- --------
------------
 City
  No. Postal
----- -------
         3        102 Crocitto, Fred  4032796411  4032449653  101-09
120th St.  NE
    2 T2J 6P3

         2        103 Landry, J.      4032794511  4032844477  7435
Boulevard East
                                                              #4  NE

    2 T2J 6B6

         8        104 Enison, Laetia  4032791223  4032557865  144-61
87th Ave  NE
    2 T2J 6B6


Mon Jan 20
page    2
                                      CUSTOMER

      Emp.       Cust
        ID         ID Name            Home        Business    ADDRESS
---------- ---------- --------------- ----------- ----------- --------
------------
 City
  No. Postal
----- -------
         2        105 Moskowitz,      4032794228  4036409874  320 John
St.  NE
                      Angel
    2 T2J 7E3

         3        106 Olvsade, Judith 4032795652  4036861598  29
Elmwood Ave.
    2 T2Z 3M9


5 rows selected.


Mon Jan 20
page    1
                                     CUSTCREDIT

Row Counts
----------
```

```
       215

1 row selected.


Mon Jan 20
page    1
                                     CUSTCREDIT

      Cust                               Card
        ID CREDITCARDNO         EXPDATE   Type
---------- -------------------- --------- ----
       104 78789007977999       22-AUG-17    1
       105 78789007977999       22-AUG-17    2
       105 3347777387984530     09-MAY-17
       106 5214635263416330     01-FEB-17
       106 3522354387984530     09-JUN-17    1

5 rows selected.


Mon Jan 20
page    1
                                     DESCRIPTION

Row Counts
----------
       278

1 row selected.


Mon Jan 20
page    1
                                     DESCRIPTION

  No. DESCRIPTION
----- ------------------------------
    1 14 NIGHTS HOTEL
    2 3NIGHTS TORONTO
    3 ADELAIDE-14 NIGHTS
    4 AFRICAN SAFARI
    5 ALASKA

5 rows selected.


Mon Jan 20
page    1
                                       INVOICE

Row Counts
```

```
----------
      1341

1 row selected.


Mon Jan 20
page    1
                                         INVOICE

Inv.
 No. ITINERARYNUM PAYMENTTYPE BILLINGDA BILLAMOUNT  REMAINDER SALEDATE
Code
---- ------------ ----------- --------- ---------- ---------- --------
- ----
                Cust.
CREDITCARD         ID
--------------- -----
   1         1000             25-FEB-14      677.2     5644.8 04-JAN-
00 BK
99834387984533    191

   2         1000             25-FEB-14     5644.8      627.2 25-JAN-
00 BK
99834387984533    191

   3         1001             06-JAN-14      627.7          0 05-JAN-
17 BK
99834387343424    221

   4         1002             05-JAN-15      627.7          0 05-JAN-
18 BK

Mon Jan 20
page    2
                                         INVOICE

Inv.
 No. ITINERARYNUM PAYMENTTYPE BILLINGDA BILLAMOUNT  REMAINDER SALEDATE
Code
---- ------------ ----------- --------- ---------- ---------- --------
- ----
                Cust.
CREDITCARD         ID
--------------- -----
12134387343424    261

   5         1003             29-JAN-14      627.7          0 29-JAN-
17 BK
24242341232433    362
```

```
5 rows selected.


Mon Jan 20
page    1
                                         LOCATION

Row Counts
----------
       146

1 row selected.


Mon Jan 20                               page    1
                     LOCATION

                          Province
 No. City                 /State
---- -------------------- ----------
Country
--------------------
   1 BANFF                AB
Canada

   2 CALGARY              AB
Canada

   3 EDMONTON             AB
Canada

   4 FORT SASKATCHEWAN    AB
Canada

Mon Jan 20                               page    2
                     LOCATION

                          Province
 No. City                 /State
---- -------------------- ----------
Country
--------------------

   5 STONY PLAIN          AB
Canada


5 rows selected.

SP2-0158: unknown COLUMN option "heaing"
```

```
Mon Jan 20
page    1
                               OPTIONALCUSTDETAIL

Row Counts
----------
       197

1 row selected.


Mon Jan 20
page    1
                               OPTIONALCUSTDETAIL

      Cust
        ID BIRTHDATE WEDDING
---------- --------- ---------
EMAIL                                              W
-------------------------------------------------- -
       102 22-JAN-70
                                                   N

       105
amoskowitz@home.com                                N

       106 22-JUN-55
jolvsade@aol.com                                   N

       107
cmierzwa@msn.com                                   N

Mon Jan 20
page    2
                               OPTIONALCUSTDETAIL

      Cust
        ID BIRTHDATE WEDDING
---------- --------- ---------
EMAIL                                              W
-------------------------------------------------- -

       108
judysehti@home.com                                 N


5 rows selected.


Mon Jan 20
page    1
                                    PRODUCT
```

```
Row Counts
----------
       198

1 row selected.


Mon Jan 20
page    1
                                    PRODUCT

   Prod. Supplier                                  Tax
Category      No. COMPANY                          No.    TAXRATE
-------- -------- ------------------------- ---------- ----------
      Base Price
----------------
     100      322 AVILA TOURS INC.
        $630.00

     100     4196 TRAVEL STUDIO
        $970.00

     100     3376 WORLD AIR TOURS
        $650.00

     100     2727 SKYLINK
        $890.00

Mon Jan 20
page    2
                                    PRODUCT

   Prod. Supplier                                  Tax
Category      No. COMPANY                          No.    TAXRATE
-------- -------- ------------------------- ---------- ----------
      Base Price
----------------

     100     2099 BALDWIN TRAVEL
        $800.00


5 rows selected.


Mon Jan 20
page    1
                                       SUPPLIER

Row Counts
----------
```

```
      1016

1 row selected.


Mon Jan 20
page    1
                                       SUPPLIER

Supplier                           Repr.
     No. Company                     No. REPRESENTS
OFFICENO Code
-------- ------------------------- ----- ------------------------- ---
------- --------
      12 21ST CENTURY TRAVEL
1 PGY
         INSURANCE

      30 DISNEY CRUISE LINE
1
      37 ACCORD TRAVEL (TOURS) INC
1 ACTAPGY
      48 SIGNATURE VACATIONS
1
      50 AER LINGUS
1

5 rows selected.


Mon Jan 20
page    1
                                       SUPPLIER

Supplier
     No. Company                   ADDRESS              Postal    No.
-------- ------------------------- -------------------- -------- ----
      12 21ST CENTURY TRAVEL       220 Duncan Mill      M3B 3J5    70
         INSURANCE                 RdSuite 620

      30 DISNEY CRUISE LINE        210 Celebration Pl   34747-4   114
      37 ACCORD TRAVEL (TOURS) INC 777 Bay StSuite 1908 M5G 2C8    70
      48 SIGNATURE VACATIONS       160 Bloor St ESuite  M5W 1B9    70
                                   400

      50 AER LINGUS                538 Broadhollow Rd   11747     135

5 rows selected.


Mon Jan 20
page    1
```

```
                                       SUPPLIER

Supplier
     No. Contact    EMAIL                     PHONE       FAX
Website
-------- ---------- ------------------------- ----------- -----------
---------------
      12 Sean       travelinsurance@globalser 4164410166  4164411241
         Russell    ve.net

      30                                      8005118444  4075667353
http://www.disn

eycruise.com

      37 Linda                                4165993340  4165993405
         Caldes

      48 Franca                               4169671510  4169677154
http://www.sign
         Iuele
ature.ca

      50                                      8002236537  6317522044

5 rows selected.


Mon Jan 20
page    1
                                     TRIPPRODUCT

Row Counts
----------
      1138

1 row selected.


Mon Jan 20
page    1
                                     TRIPPRODUCT

                        Supplier                                  Prod.
    LISTNO ITINERARYNUM      No. COMPANY                       Category
BOOKINGNUM
---------- ------------ -------- ------------------------- -------- --
-------------
         1         1000     6346 ELDERTREKS                     501
56565T
         3         1002     3212 UNITED AIRLINES                150
SDF24896
```

```
         5         1004     3212 UNITED AIRLINES                 150
LJJ98
         6         1005     3212 UNITED AIRLINES                 150
SDF24899
         7         1006     3212 UNITED AIRLINES                 150
SDF24900

5 rows selected.


Mon Jan 20
page    1
                                     TRIPPRODUCT

                               Dest        Desc.
    LISTNO STARTDATE ENDDATE   Code  Class   No. Destination
---------- --------- --------- ----- ----- ----- ---------------
         1 25-MAY-17 22-JUN-17 EU              197 Europe
         3 15-FEB-18 20-FEB-18 NA    BSN     114 Winnipeg
         5 14-MAY-17 19-MAY-17 NA    BSN      88 Hamilton
         6 15-FEB-18 20-FEB-18 NA    BSN     114 Winnipeg
         7 21-FEB-17 26-FEB-17 SA    BSN      77 Buenos Aires

5 rows selected.


Mon Jan 20                                         page    1
                            TRIP

Row Counts
----------
       971

1 row selected.


Mon Jan 20                                         page    1
                            TRIP

Itinerary             Cust Trip      Trip
      No. Code          ID Start     End       TRAVELLERS
--------- ----- ---------- --------- --------- ----------
     1000              191 25-MAY-17 22-JUN-17          2
     1001              221 20-FEB-17 25-FEB-17          1
     1002              261 15-FEB-18 20-FEB-18          1
     1003              362 20-FEB-17 25-FEB-17          1
     1004              379 14-MAY-17 19-MAY-17          1

5 rows selected.


Mon Jan 20                                         page    1
```

```
                       TRIPTRAVELLER

Itinerary
      No. Traveller
--------- --------------------
      584 John Doe

1 row selected.


Mon Jan 20                                         page    1
                        Reward Cards

Row Counts
----------
         5

1 row selected.


Mon Jan 20                                         page    1
                        Reward Cards

 No. Points Program
---- --------------------
   1 WestJet Rewards
   2 Airmiles
   3 SkyMiles
   4 Aeroplan
   5 MileagePlus

5 rows selected.


Mon Jan 20                                         page    1
                   Customer Rewards Cards

Row Counts
----------
         2

1 row selected.


Mon Jan 20                                         page    1
                   Customer Rewards Cards

      Cust
        ID CARDNO               Points Program
---------- -------------------- --------------------
       140 112202               Airmiles
       347 110202               WestJet Rewards
```

```
2 rows selected.


Mon Jan 20                                         page    1
                       Payment Types

Row Counts
----------
         4

1 row selected.


Mon Jan 20                                         page    1
                       Payment Types

 No. Description
---- --------------------
   1 Full Payment
   2 Payment
   3 Final Payment
   4 Deposit

4 rows selected.


Mon Jan 20                                         page    1
                         Tax Types

Row Counts
----------
         4

1 row selected.


Mon Jan 20                                         page    1
                         Tax Types

 No. Description
---- --------------------
   1 PST Only
   2 GST Only
   3 GST + PST
   4 Environmental Tax

4 rows selected.

USER is "THEDBA"

Mon Jan 20                     page    1
```

```
                 Verify

Database name:  TEA
Instance name:  tea

Date:  20-JAN-20
USER is "THEDBA"
SP2-0158: unknown COLUMN option "a7"

Mon Jan 20                                                 page    1
                                Files

STATUS  NAME
------- --------------------------------------------------
        C:\TEA\DISK1\REDO01A.LOG
        C:\TEA\DISK1\REDO02A.LOG
        C:\TEA\DISK1\REDO03A.LOG
        C:\TEA\DISK2\REDO01B.LOG
        C:\TEA\DISK2\REDO02B.LOG
        C:\TEA\DISK2\REDO03B.LOG
        C:\TEA\DISK3\REDO01C.LOG
        C:\TEA\DISK3\REDO02C.LOG
        C:\TEA\DISK3\REDO03C.LOG
SYSTEM  C:\TEA\DISK1\SYSTEM01.DBF
ONLINE  C:\TEA\DISK1\RO.DBF
ONLINE  C:\TEA\DISK2\INDX.DBF
ONLINE  C:\TEA\DISK3\ADHOC.DBF
ONLINE  C:\TEA\DISK3\UNDO01.DBF
ONLINE  C:\TEA\DISK3\USER_DATA01.DBF
ONLINE  C:\TEA\DISK4\SYSAUX01.DBF
ONLINE  C:\TEA\DISK4\UNDO02.DBF
ONLINE  C:\TEA\DISK4\USER_DATA02.DBF
Control C:\TEA\DISK1\CONTROL01.CTL
Control C:\TEA\DISK2\CONTROL02.CTL
Control C:\TEA\DISK3\CONTROL03.CTL
Control C:\TEA\DISK5\CONTROL04.CTL
SP2-0735: unknown COLUMN option beginning "WORD_WRAPP..."

Mon Jan 20                                                 page    1
                              Log Files

NAME
--------------------------------------------------
VALUE
--------------------------------------------------
Active Problem Count
0

Active Incident Count
0

ADR Base
```

```
C:\TEA\DISK5

ADR Home
C:\TEA\DISK5\diag\rdbms\tea\tea

Diag Alert
C:\TEA\DISK5\diag\rdbms\tea\tea\alert

Diag Cdump
C:\TEA\DISK5\diag\rdbms\tea\tea\cdump

Health Monitor
C:\TEA\DISK5\diag\rdbms\tea\tea\hm

Diag Incident
C:\TEA\DISK5\diag\rdbms\tea\tea\incident

Diag Trace
C:\TEA\DISK5\diag\rdbms\tea\tea\trace

Default Trace File
C:\TEA\DISK5\diag\rdbms\tea\tea\trace\tea_ora_6008
.trc

Diag Enabled

Mon Jan 20                                                   page    2
                              Log Files

NAME
--------------------------------------------------
VALUE
--------------------------------------------------
TRUE


Mon Jan 20                                                   page    1
                              Log Files

    GROUP# STATUS              MEMBERS
---------- ---------------- ----------
         1 ACTIVE                    3
         2 CURRENT                   3
         3 ACTIVE                    3

Mon Jan 20                                                   page    1
                             TableSpaces

 TS# Loc                                                  TS Name
---- ---------------------------------------------------- ----------
   0 C:\TEA\DISK1\SYSTEM01.DBF                            SYSTEM
   1 C:\TEA\DISK4\SYSAUX01.DBF                            SYSAUX
   2 C:\TEA\DISK3\UNDO01.DBF                                 UNDOTBS01
   4 C:\TEA\DISK3\USER_DATA01.DBF                            USERDATA01
   5 C:\TEA\DISK4\USER_DATA02.DBF                            USERDATA02
   6 C:\TEA\DISK4\UNDO02.DBF                                 UNDOTBS02
   7 C:\TEA\DISK2\INDX.DBF                                   INDX
   8 C:\TEA\DISK3\ADHOC.DBF                                  ADHOC
   9 C:\TEA\DISK1\RO.DBF                                     RO

Mon Jan 20                                                   page    1
                             TableSpaces

TS Name              STATUS    LOGGING   RETENTION   Block Size
-------------------- --------- --------- ----------- ----------
ADHOC                ONLINE    LOGGING   NOT APPLY         8192
INDX                 ONLINE    LOGGING   NOT APPLY         8192
RO                   ONLINE    LOGGING   NOT APPLY         8192
SYSAUX               ONLINE    LOGGING   NOT APPLY         8192
SYSTEM               ONLINE    LOGGING   NOT APPLY         8192
TEMP01               ONLINE    NOLOGGING NOT APPLY         8192
UNDOTBS01            ONLINE    LOGGING   NOGUARANTEE       8192
UNDOTBS02            ONLINE    LOGGING   NOGUARANTEE       8192
USERDATA01           ONLINE    LOGGING   NOT APPLY         8192
USERDATA02           ONLINE    LOGGING   NOT APPLY         8192

Mon Jan 20                     page    1
                  SGA

NAME                                VALUE
------------------------- --------------
Fixed Size                4629240
Variable Size             1040190728
Database Buffers          4110417920
Redo Buffers              12144640

Mon Jan 20                                                   page    1
                              Parameters

NAME                           VALUE
------------------------------ -----------------------------------
audit_trail                    DB, EXTENDED
compatible                     12.0.0
db_block_size                  8192
db_name                        TEA
db_recovery_file_dest          C:\TEA\DISK5
db_recovery_file_dest_size     10009706496
diagnostic_dest                C:\TEA\DISK5
instance_name                  TEA
processes                      150
sga_target                     5167382528
undo_management                AUTO
undo_retention                 1800
undo_tablespace                UNDOTBS01
```

```
SQL> TTITLE OFF
SQL> BTITLE OFF
SQL> SET linesize 60
SQL> SET pagesize 60
SQL> SET long 80
SQL> spool off
```

Workshop 5

```
SQL>
SQL> COL USERNAME FOR A20
SQL> COL STATUS FOR A20
SQL> COL 'DEFAULT TS' FOR A15
SQL> COL 'TMP TS' FOR A8
SQL> COL 'LAST LOG' FOR A12
SQL> col profile for a15
SQL> SET LINESIZE 100
SQL> SET PAGESIZE 50
SQL> TTITLE 'Users'
SQL> SELECT USERNAME, ACCOUNT_STATUS STATUS,
  2   DEFAULT_TABLESPACE "DEFAULT TS", TEMPORARY_TABLESPACE "TMP TS",
to_char(LAST_LOGIN,'MON DD, YYYY') "LAST LOG", profile
  3  FROM DBA_USERS
  4  ORDER BY 2 , 1;

Mon Jan 20
page    1
                                               Users

USERNAME             STATUS               DEFAULT TS      TMP TS
LAST LOG     PROFILE
-------------------- -------------------- --------------- -------- ---
--------- ---------------
COM_SPECIAL          EXPIRED              USERDATA02      TEMP01
MANAGER
INT_AGENT            EXPIRED              USERDATA02      TEMP01
INT_AGENT
JR_AGENT             EXPIRED              USERDATA02      TEMP01
JR_AGENT
MANAGER              EXPIRED              USERDATA02      TEMP01
MANAGER
OWNER                EXPIRED              USERDATA02      TEMP01
OWNER
SR_AGENT             EXPIRED              USERDATA02      TEMP01
SR_AGENT
SUPPORT              EXPIRED              USERDATA02      TEMP01
INT_AGENT
APPQOSSYS            EXPIRED & LOCKED     SYSAUX          TEMP01
DEFAULT
AUDSYS               EXPIRED & LOCKED     SYSTEM          TEMP01
DBSNMP               EXPIRED & LOCKED     SYSAUX          TEMP01
```

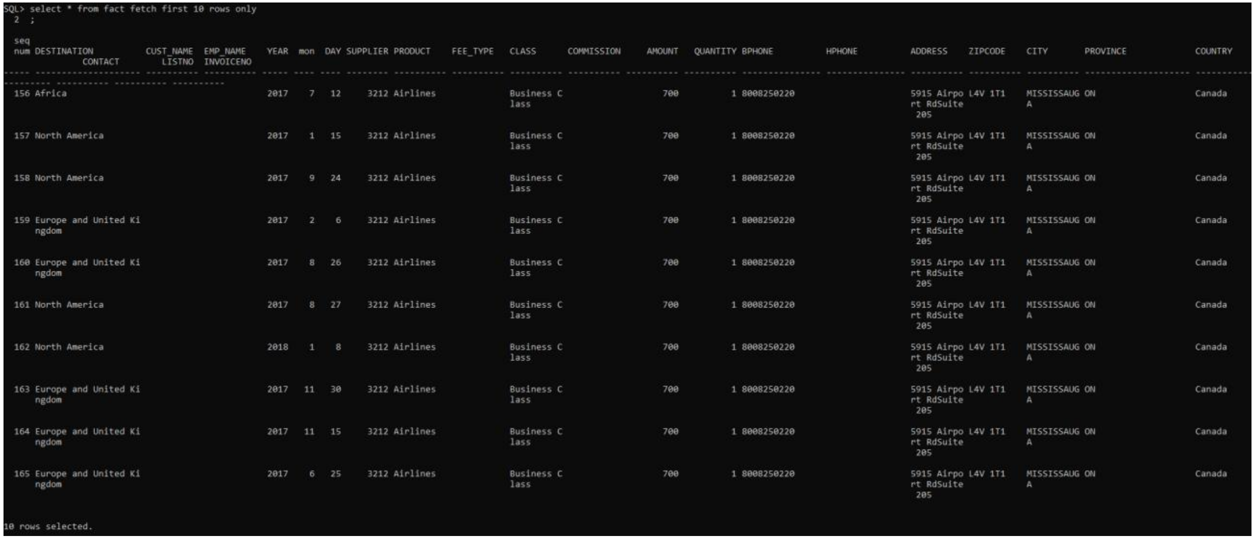
```
DIP                  EXPIRED & LOCKED     SYSTEM          TEMP01
GSMADMIN_INTERNAL    EXPIRED & LOCKED     SYSAUX          TEMP01
GSMCATUSER           EXPIRED & LOCKED     SYSTEM          TEMP01
GSMUSER              EXPIRED & LOCKED     SYSTEM          TEMP01
ORACLE_OCM           EXPIRED & LOCKED     SYSTEM          TEMP01
OUTLN                EXPIRED & LOCKED     SYSTEM          TEMP01
SYSBACKUP            EXPIRED & LOCKED     SYSTEM          TEMP01
SYSDG                EXPIRED & LOCKED     SYSTEM          TEMP01
SYSKM                EXPIRED & LOCKED     SYSTEM          TEMP01
XDB                  EXPIRED & LOCKED     SYSAUX          TEMP01
XS$NULL              EXPIRED & LOCKED     SYSTEM          TEMP01
ANONYMOUS            LOCKED               SYSAUX          TEMP01
DONOTTOUCH           OPEN                 USERDATA01      TEMP01   JAN
20, 2020 DBA
SYS                  OPEN                 SYSTEM          TEMP01
DEFAULT
SYSTEM               OPEN                 SYSTEM          TEMP01   JAN
08, 2020
THEDBA               OPEN                 USERDATA01      TEMP01   JAN
20, 2020 DBA

26 rows selected.

SQL>
SQL>
SQL> TTITLE 'PRIVILEGES'
SQL> col grantee for a15
SQL> col owner for a10
SQL> col table_name for a15 heading "Table" word_wrapped
SQL> col privilege for a15
SQL> select grantee, owner, table_name, privilege
  2  FROM dba_tab_privs
  3  where OWNER = 'DONOTTOUCH'
  4  order by 1,3;

Mon Jan 20
page    1
                                             PRIVILEGES

GRANTEE         OWNER      Table           PRIVILEGE
--------------- ---------- --------------- ---------------
COM_SPECIAL_R   DONOTTOUCH COMMISSION      DELETE
COM_SPECIAL_R   DONOTTOUCH COMMISSION      UPDATE
COM_SPECIAL_R   DONOTTOUCH COMMISSION      INSERT
COM_SPECIAL_R   DONOTTOUCH COMMISSION      SELECT
INT_AGENT_R     DONOTTOUCH CREDITCARD      SELECT
INT_AGENT_R     DONOTTOUCH CUSTCREDIT      SELECT
INT_AGENT_R     DONOTTOUCH CUSTREWARDCARD  UPDATE
INT_AGENT_R     DONOTTOUCH DESCRIPTION     INSERT
INT_AGENT_R     DONOTTOUCH INVOICE         INSERT
INT_AGENT_R     DONOTTOUCH INVOICE         SELECT
INT_AGENT_R     DONOTTOUCH PAYMENTTYPE     SELECT
INT_AGENT_R     DONOTTOUCH REWARDCARD      INSERT
INT_AGENT_R     DONOTTOUCH TRIP            UPDATE
INT_AGENT_R     DONOTTOUCH TRIPPRODUCT     UPDATE
JR_AGENT_R      DONOTTOUCH CLASS           SELECT
JR_AGENT_R      DONOTTOUCH COMMISSION      INSERT
JR_AGENT_R      DONOTTOUCH COMMISSIONTYPE  SELECT
JR_AGENT_R      DONOTTOUCH CUSTCREDIT      INSERT
JR_AGENT_R      DONOTTOUCH CUSTOMER        INSERT
JR_AGENT_R      DONOTTOUCH CUSTOMER        UPDATE
JR_AGENT_R      DONOTTOUCH CUSTREWARDCARD  INSERT
JR_AGENT_R      DONOTTOUCH CUSTREWARDCARD  SELECT
JR_AGENT_R      DONOTTOUCH DESCRIPTION     SELECT
JR_AGENT_R      DONOTTOUCH DESTINATION     SELECT
JR_AGENT_R      DONOTTOUCH FEETYPE         SELECT
JR_AGENT_R      DONOTTOUCH LOCATION        SELECT
JR_AGENT_R      DONOTTOUCH OPTIONALCUSTDET INSERT
                           AIL

JR_AGENT_R      DONOTTOUCH OPTIONALCUSTDET UPDATE
                           AIL

JR_AGENT_R      DONOTTOUCH PRODUCT         SELECT
JR_AGENT_R      DONOTTOUCH PRODUCTCAT      SELECT
JR_AGENT_R      DONOTTOUCH REWARDCARD      SELECT
JR_AGENT_R      DONOTTOUCH SUPPLIER        SELECT
JR_AGENT_R      DONOTTOUCH TAXTYPE         SELECT
JR_AGENT_R      DONOTTOUCH TRIP            INSERT
JR_AGENT_R      DONOTTOUCH TRIP            SELECT
JR_AGENT_R      DONOTTOUCH TRIPPRODUCT     SELECT
JR_AGENT_R      DONOTTOUCH TRIPPRODUCT     INSERT
JR_AGENT_R      DONOTTOUCH TRIPTRAVELLER   DELETE
JR_AGENT_R      DONOTTOUCH TRIPTRAVELLER   INSERT
JR_AGENT_R      DONOTTOUCH TRIPTRAVELLER   SELECT

Mon Jan 20
page    2
                                             PRIVILEGES

GRANTEE         OWNER      Table           PRIVILEGE
--------------- ---------- --------------- ---------------
JR_AGENT_R      DONOTTOUCH TRIPTRAVELLER   UPDATE
MANAGER_R       DONOTTOUCH COMMISSION      INSERT
MANAGER_R       DONOTTOUCH COMMISSION      UPDATE
MANAGER_R       DONOTTOUCH COMMISSION      DELETE
MANAGER_R       DONOTTOUCH DESTINATION     DELETE
MANAGER_R       DONOTTOUCH DESTINATION     UPDATE
MANAGER_R       DONOTTOUCH DESTINATION     INSERT
MANAGER_R       DONOTTOUCH EMPLOYEE        DELETE
MANAGER_R       DONOTTOUCH EMPLOYEE        SELECT
MANAGER_R       DONOTTOUCH EMPLOYEE        INSERT
MANAGER_R       DONOTTOUCH EMPLOYEE        UPDATE
MANAGER_R       DONOTTOUCH LOCATION        INSERT
MANAGER_R       DONOTTOUCH LOCATION        DELETE
MANAGER_R       DONOTTOUCH LOCATION        UPDATE
MANAGER_R       DONOTTOUCH PRODUCT         INSERT
MANAGER_R       DONOTTOUCH PRODUCT         DELETE
MANAGER_R       DONOTTOUCH PRODUCT         UPDATE
MANAGER_R       DONOTTOUCH SUPPLIER        INSERT
MANAGER_R       DONOTTOUCH SUPPLIER        DELETE
MANAGER_R       DONOTTOUCH SUPPLIER        UPDATE
MANAGER_R       DONOTTOUCH TAXTYPE         DELETE
MANAGER_R       DONOTTOUCH TAXTYPE         UPDATE
MANAGER_R       DONOTTOUCH TAXTYPE         INSERT
OWNER           DONOTTOUCH AFFILIATION     SELECT
OWNER           DONOTTOUCH CLASS           SELECT
OWNER           DONOTTOUCH COMMISSION      SELECT
OWNER           DONOTTOUCH COMMISSIONTYPE  SELECT
OWNER           DONOTTOUCH CREDITCARD      SELECT
OWNER           DONOTTOUCH CUSTCREDIT      SELECT
OWNER           DONOTTOUCH CUSTOMER        SELECT
OWNER           DONOTTOUCH CUSTREWARDCARD  SELECT
OWNER           DONOTTOUCH DESCRIPTION     SELECT
OWNER           DONOTTOUCH DESTINATION     SELECT
OWNER           DONOTTOUCH EMPLOYEE        SELECT
OWNER           DONOTTOUCH FEETYPE         SELECT
OWNER           DONOTTOUCH INVOICE         SELECT
OWNER           DONOTTOUCH LOCATION        SELECT
OWNER           DONOTTOUCH OPTIONALCUSTDET SELECT
                           AIL

OWNER           DONOTTOUCH PAYMENTTYPE     SELECT
OWNER           DONOTTOUCH PRODUCT         SELECT
OWNER           DONOTTOUCH PRODUCTCAT      SELECT
OWNER           DONOTTOUCH REWARDCARD      SELECT

Mon Jan 20
page    3
                                             PRIVILEGES

GRANTEE         OWNER      Table           PRIVILEGE
--------------- ---------- --------------- ---------------
OWNER           DONOTTOUCH SUPPLIER        SELECT
OWNER           DONOTTOUCH TAXTYPE         SELECT
OWNER           DONOTTOUCH TRIP            SELECT
OWNER           DONOTTOUCH TRIPPRODUCT     SELECT
OWNER           DONOTTOUCH TRIPTRAVELLER   SELECT
OWNER           DONOTTOUCH TRIPTYPE        SELECT
SR_AGENT_R      DONOTTOUCH CUSTCREDIT      UPDATE
SR_AGENT_R      DONOTTOUCH INVOICE         UPDATE
SR_AGENT_R      DONOTTOUCH PRODUCT         UPDATE
SR_AGENT_R      DONOTTOUCH SUPPLIER        UPDATE
SUPPORT_R       DONOTTOUCH CUSTOMER        SELECT
SUPPORT_R       DONOTTOUCH CUSTREWARDCARD  SELECT
SUPPORT_R       DONOTTOUCH EMP_VIEW        SELECT
SUPPORT_R       DONOTTOUCH OPTIONALCUSTDET SELECT
                           AIL


96 rows selected.

SQL>
SQL>
SQL> TTITLE 'Granted Roles'
SQL> COL GRANTED_ROLE FOR A20
SQL> select grantee, granted_role
  2  from dba_role_privs
  3  where grantee IN ('JR_AGENT', 'SR_AGENT','OWNER','INT_AGENT',
'SUPPORT', 'MANAGER','COM_SPECIAL');

Mon Jan 20
page    1
                                           Granted Roles

GRANTEE         GRANTED_ROLE
--------------- --------------------
INT_AGENT       INT_AGENT_R
COM_SPECIAL     COM_SPECIAL_R
OWNER           OWNER_R
JR_AGENT        JR_AGENT_R
SUPPORT         SUPPORT_R
MANAGER         MANAGER_R
SR_AGENT        SR_AGENT_R

7 rows selected.

SQL>
SQL>
SQL> ttitle 'Profiles'
SQL> col profile for a20 word_wrapped
SQL> col resource_name for a32 heading "Resource" word_wrapped
SQL> col limit for a10 word_wrapped
SQL> col resource_type for a8
SQL> break on profile
SQL> select profile, resource_name, resource_type, limit
  2  FROM dba_profiles
  3  order by 1;

Mon Jan 20
page    1
                                              Profiles

PROFILE              Resource                         RESOURCE LIMIT
-------------------- -------------------------------- -------- -------
---
DBA                  CPU_PER_SESSION                  KERNEL   3000
                     COMPOSITE_LIMIT                  KERNEL   DEFAULT
                     SESSIONS_PER_USER                KERNEL   5
                     CPU_PER_CALL                     KERNEL   6000
                     IDLE_TIME                        KERNEL
UNLIMITED
                     PASSWORD_GRACE_TIME              PASSWORD DEFAULT
                     PASSWORD_VERIFY_FUNCTION         PASSWORD DEFAULT
                     PASSWORD_REUSE_MAX               PASSWORD DEFAULT
                     PASSWORD_REUSE_TIME              PASSWORD DEFAULT
                     PASSWORD_LIFE_TIME               PASSWORD DEFAULT
                     FAILED_LOGIN_ATTEMPTS            PASSWORD DEFAULT
                     PRIVATE_SGA                      KERNEL   DEFAULT
                     CONNECT_TIME                     KERNEL   DEFAULT
                     LOGICAL_READS_PER_CALL           KERNEL   DEFAULT
                     LOGICAL_READS_PER_SESSION        KERNEL   DEFAULT
                     PASSWORD_LOCK_TIME               PASSWORD .0208
DEFAULT              CONNECT_TIME                     KERNEL   480
                     PASSWORD_LOCK_TIME               PASSWORD .0416
                     FAILED_LOGIN_ATTEMPTS            PASSWORD 5
                     IDLE_TIME                        KERNEL   5
                     PASSWORD_LIFE_TIME               PASSWORD 90
                     PASSWORD_GRACE_TIME              PASSWORD 7
                     PASSWORD_VERIFY_FUNCTION         PASSWORD NULL
                     PASSWORD_REUSE_MAX               PASSWORD
UNLIMITED
                     PASSWORD_REUSE_TIME              PASSWORD
UNLIMITED
                     PRIVATE_SGA                      KERNEL
UNLIMITED
                     LOGICAL_READS_PER_CALL           KERNEL
UNLIMITED
                     LOGICAL_READS_PER_SESSION        KERNEL
UNLIMITED
                     CPU_PER_CALL                     KERNEL
UNLIMITED
                     CPU_PER_SESSION                  KERNEL
UNLIMITED
                     SESSIONS_PER_USER                KERNEL
UNLIMITED
                     COMPOSITE_LIMIT                  KERNEL
UNLIMITED
INT_AGENT            CPU_PER_SESSION                  KERNEL   3000
                     COMPOSITE_LIMIT                  KERNEL   DEFAULT
                     SESSIONS_PER_USER                KERNEL   2
                     IDLE_TIME                        KERNEL   10
                     PASSWORD_GRACE_TIME              PASSWORD DEFAULT
                     PASSWORD_LOCK_TIME               PASSWORD DEFAULT
                     PASSWORD_VERIFY_FUNCTION         PASSWORD DEFAULT
                     PASSWORD_REUSE_MAX               PASSWORD DEFAULT
                     PASSWORD_REUSE_TIME              PASSWORD DEFAULT
                     PASSWORD_LIFE_TIME               PASSWORD DEFAULT
                     FAILED_LOGIN_ATTEMPTS            PASSWORD DEFAULT
                     PRIVATE_SGA                      KERNEL   DEFAULT
```

```
Mon Jan 20
page    2
                                              Profiles

PROFILE              Resource                         RESOURCE LIMIT
-------------------- -------------------------------- -------- -------
---
INT_AGENT            CONNECT_TIME                     KERNEL   DEFAULT
                     LOGICAL_READS_PER_CALL           KERNEL   DEFAULT
                     LOGICAL_READS_PER_SESSION        KERNEL   DEFAULT
                     CPU_PER_CALL                     KERNEL   300
JR_AGENT             CPU_PER_CALL                     KERNEL   3000
                     COMPOSITE_LIMIT                  KERNEL   DEFAULT
                     SESSIONS_PER_USER                KERNEL   2
                     IDLE_TIME                        KERNEL   10
                     PASSWORD_GRACE_TIME              PASSWORD DEFAULT
                     PASSWORD_LOCK_TIME               PASSWORD DEFAULT
                     PASSWORD_VERIFY_FUNCTION         PASSWORD DEFAULT
                     PASSWORD_REUSE_MAX               PASSWORD DEFAULT
                     PASSWORD_REUSE_TIME              PASSWORD DEFAULT
                     PASSWORD_LIFE_TIME               PASSWORD DEFAULT
                     FAILED_LOGIN_ATTEMPTS            PASSWORD DEFAULT
                     PRIVATE_SGA                      KERNEL   DEFAULT
                     CONNECT_TIME                     KERNEL   DEFAULT
                     LOGICAL_READS_PER_CALL           KERNEL   DEFAULT
                     LOGICAL_READS_PER_SESSION        KERNEL   DEFAULT
                     CPU_PER_SESSION                  KERNEL   3000
MANAGER              SESSIONS_PER_USER                KERNEL   3
                     COMPOSITE_LIMIT                  KERNEL   DEFAULT
                     CPU_PER_CALL                     KERNEL   6000
                     IDLE_TIME                        KERNEL   10
                     PASSWORD_GRACE_TIME              PASSWORD DEFAULT
                     PASSWORD_LOCK_TIME               PASSWORD DEFAULT
                     PASSWORD_VERIFY_FUNCTION         PASSWORD DEFAULT
                     PASSWORD_REUSE_MAX               PASSWORD DEFAULT
                     PASSWORD_REUSE_TIME              PASSWORD DEFAULT
                     PASSWORD_LIFE_TIME               PASSWORD DEFAULT
                     FAILED_LOGIN_ATTEMPTS            PASSWORD DEFAULT
                     PRIVATE_SGA                      KERNEL   DEFAULT
                     CONNECT_TIME                     KERNEL   DEFAULT
                     LOGICAL_READS_PER_CALL           KERNEL   DEFAULT
                     LOGICAL_READS_PER_SESSION        KERNEL   DEFAULT
                     CPU_PER_SESSION                  KERNEL   3000
ORA_STIG_PROFILE     SESSIONS_PER_USER                KERNEL   DEFAULT
                     COMPOSITE_LIMIT                  KERNEL   DEFAULT
                     CPU_PER_CALL                     KERNEL   DEFAULT
                     LOGICAL_READS_PER_SESSION        KERNEL   DEFAULT
                     LOGICAL_READS_PER_CALL           KERNEL   DEFAULT
                     CONNECT_TIME                     KERNEL   DEFAULT
                     PRIVATE_SGA                      KERNEL   DEFAULT
                     IDLE_TIME                        KERNEL   15
```

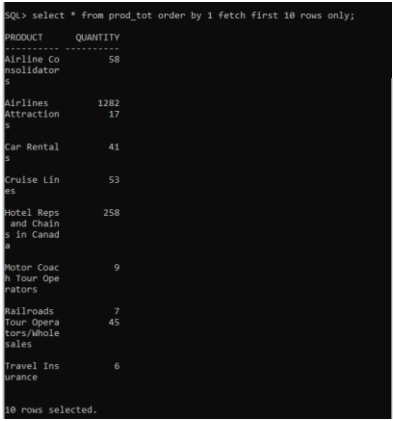
```
Mon Jan 20
page    3
                                              Profiles

PROFILE              Resource                         RESOURCE LIMIT
-------------------- -------------------------------- -------- -------
---
ORA_STIG_PROFILE     PASSWORD_LOCK_TIME               PASSWORD
UNLIMITED
                     FAILED_LOGIN_ATTEMPTS            PASSWORD 3
                     PASSWORD_GRACE_TIME              PASSWORD 5
                     PASSWORD_LIFE_TIME               PASSWORD 60
                     PASSWORD_REUSE_TIME              PASSWORD 365
                     PASSWORD_VERIFY_FUNCTION         PASSWORD
ORA12C_STR

ONG_VERIFY

_FUNCTION

                     PASSWORD_REUSE_MAX               PASSWORD 10
                     CPU_PER_SESSION                  KERNEL   DEFAULT
OWNER                COMPOSITE_LIMIT                  KERNEL   DEFAULT
                     LOGICAL_READS_PER_SESSION        KERNEL   DEFAULT
                     LOGICAL_READS_PER_CALL           KERNEL   DEFAULT
                     CONNECT_TIME                     KERNEL   DEFAULT
                     PRIVATE_SGA                      KERNEL   DEFAULT
                     FAILED_LOGIN_ATTEMPTS            PASSWORD DEFAULT
                     PASSWORD_REUSE_TIME              PASSWORD DEFAULT
                     PASSWORD_REUSE_MAX               PASSWORD DEFAULT
                     PASSWORD_VERIFY_FUNCTION         PASSWORD DEFAULT
                     PASSWORD_GRACE_TIME              PASSWORD DEFAULT
                     PASSWORD_LIFE_TIME               PASSWORD 200
                     SESSIONS_PER_USER                KERNEL   2
                     IDLE_TIME                        KERNEL   15
                     PASSWORD_LOCK_TIME               PASSWORD .0208
                     CPU_PER_SESSION                  KERNEL   3000
                     CPU_PER_CALL                     KERNEL   3000
SR_AGENT             CPU_PER_SESSION                  KERNEL   3000
                     COMPOSITE_LIMIT                  KERNEL   DEFAULT
                     IDLE_TIME                        KERNEL   15
                     SESSIONS_PER_USER                KERNEL   2
                     PASSWORD_GRACE_TIME              PASSWORD DEFAULT
                     PASSWORD_LOCK_TIME               PASSWORD DEFAULT
                     PASSWORD_VERIFY_FUNCTION         PASSWORD DEFAULT
                     PASSWORD_REUSE_MAX               PASSWORD DEFAULT
                     PASSWORD_REUSE_TIME              PASSWORD DEFAULT
                     PASSWORD_LIFE_TIME               PASSWORD DEFAULT
                     FAILED_LOGIN_ATTEMPTS            PASSWORD DEFAULT
                     PRIVATE_SGA                      KERNEL   DEFAULT
                     CONNECT_TIME                     KERNEL   DEFAULT
```

```
                         LOGICAL_READS_PER_CALL          KERNEL    DEFAULT
                         LOGICAL_READS_PER_SESSION       KERNEL    DEFAULT
                         CPU_PER_CALL                    KERNEL    6000

128 rows selected.

SQL> spool off;
```

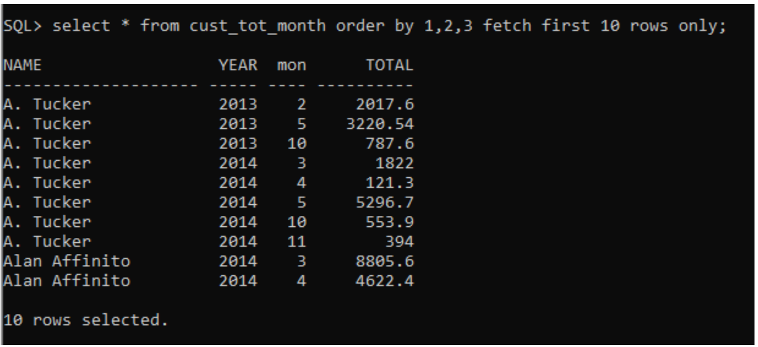
Workshop 7



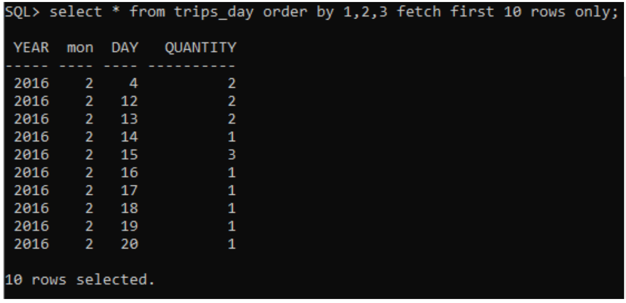
```
SQL> select * from fact fetch first 10 rows only
  2  ;

 seq
 num DESTINATION          CUST_NAME  EMP_NAME   YEAR  mon  DAY SUPPLIER PRODUCT    FEE_TYPE   CLASS      COMMISSION     AMOUNT   QUANTITY BPHONE         HPHONE         ADDRESS    ZIPCODE    CITY       PROVINCE             COUNTRY
             CONTACT          LISTNO INVOICENO
----- -------------------- ---------- ---------- ----- ---- ---- -------- ---------- ---------- ---------- ---------- ---------- ---------- -------------- -------------- ---------- ---------- ---------- -------------------- ----------
--------- ---------- ---------- ----------
  156 Africa                                      2017    7   12     3212 Airlines              Business C                   700          1 8008250220                    5915 Airpo L4V 1T1    MISSISSAUG ON                   Canada
                                                                                                lass                                                                        rt RdSuite            A
                                                                                                                                                                             205

  157 North America                               2017    1   15     3212 Airlines              Business C                   700          1 8008250220                    5915 Airpo L4V 1T1    MISSISSAUG ON                   Canada
                                                                                                lass                                                                        rt RdSuite            A
                                                                                                                                                                             205

  158 North America                               2017    9   24     3212 Airlines              Business C                   700          1 8008250220                    5915 Airpo L4V 1T1    MISSISSAUG ON                   Canada
                                                                                                lass                                                                        rt RdSuite            A
                                                                                                                                                                             205

  159 Europe and United Ki                        2017    2    6     3212 Airlines              Business C                   700          1 8008250220                    5915 Airpo L4V 1T1    MISSISSAUG ON                   Canada
      ngdom                                                                                     lass                                                                        rt RdSuite            A
                                                                                                                                                                             205

  160 Europe and United Ki                        2017    8   26     3212 Airlines              Business C                   700          1 8008250220                    5915 Airpo L4V 1T1    MISSISSAUG ON                   Canada
      ngdom                                                                                     lass                                                                        rt RdSuite            A
                                                                                                                                                                             205

  161 North America                               2017    8   27     3212 Airlines              Business C                   700          1 8008250220                    5915 Airpo L4V 1T1    MISSISSAUG ON                   Canada
                                                                                                lass                                                                        rt RdSuite            A
                                                                                                                                                                             205

  162 North America                               2018    1    8     3212 Airlines              Business C                   700          1 8008250220                    5915 Airpo L4V 1T1    MISSISSAUG ON                   Canada
                                                                                                lass                                                                        rt RdSuite            A
                                                                                                                                                                             205

  163 Europe and United Ki                        2017   11   30     3212 Airlines              Business C                   700          1 8008250220                    5915 Airpo L4V 1T1    MISSISSAUG ON                   Canada
      ngdom                                                                                     lass                                                                        rt RdSuite            A
                                                                                                                                                                             205

  164 Europe and United Ki                        2017   11   15     3212 Airlines              Business C                   700          1 8008250220                    5915 Airpo L4V 1T1    MISSISSAUG ON                   Canada
      ngdom                                                                                     lass                                                                        rt RdSuite            A
                                                                                                                                                                             205

  165 Europe and United Ki                        2017    6   25     3212 Airlines              Business C                   700          1 8008250220                    5915 Airpo L4V 1T1    MISSISSAUG ON                   Canada
      ngdom                                                                                     lass                                                                        rt RdSuite            A
                                                                                                                                                                             205


10 rows selected.
```



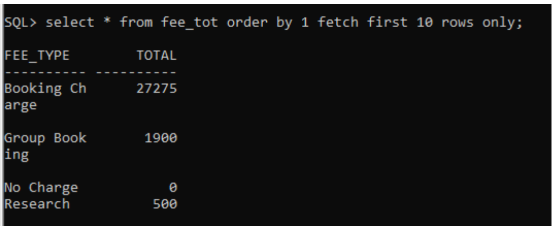
```
SQL> select * from prod_tot order by 1 fetch first 10 rows only;

PRODUCT      QUANTITY
---------- ----------
Airline Co         58
nsolidator
s

Airlines         1282
Attraction         17
s

Car Rental         41
s

Cruise Lin         53
es

Hotel Reps        258
 and Chain
s in Canad
a

Motor Coac          9
h Tour Ope
rators

Railroads           7
Tour Opera         45
tors/Whole
sales

Travel Ins          6
urance


10 rows selected.
```



```
SQL> select * from cust_tot_month order by 1,2,3 fetch first 10 rows only;

NAME                  YEAR  mon      TOTAL
-------------------- ----- ---- ----------
A. Tucker             2013    2     2017.6
A. Tucker             2013    5    3220.54
A. Tucker             2013   10      787.6
A. Tucker             2014    3       1822
A. Tucker             2014    4      121.3
A. Tucker             2014    5     5296.7
A. Tucker             2014   10      553.9
A. Tucker             2014   11        394
Alan Affinito         2014    3     8805.6
Alan Affinito         2014    4     4622.4

10 rows selected.
```



```
SQL> select * from trips_day order by 1,2,3 fetch first 10 rows only;

 YEAR  mon  DAY   QUANTITY
----- ---- ---- ----------
 2016    2    4          2
 2016    2   12          2
 2016    2   13          2
 2016    2   14          1
 2016    2   15          3
 2016    2   16          1
 2016    2   17          1
 2016    2   18          1
 2016    2   19          1
 2016    2   20          1

10 rows selected.
```



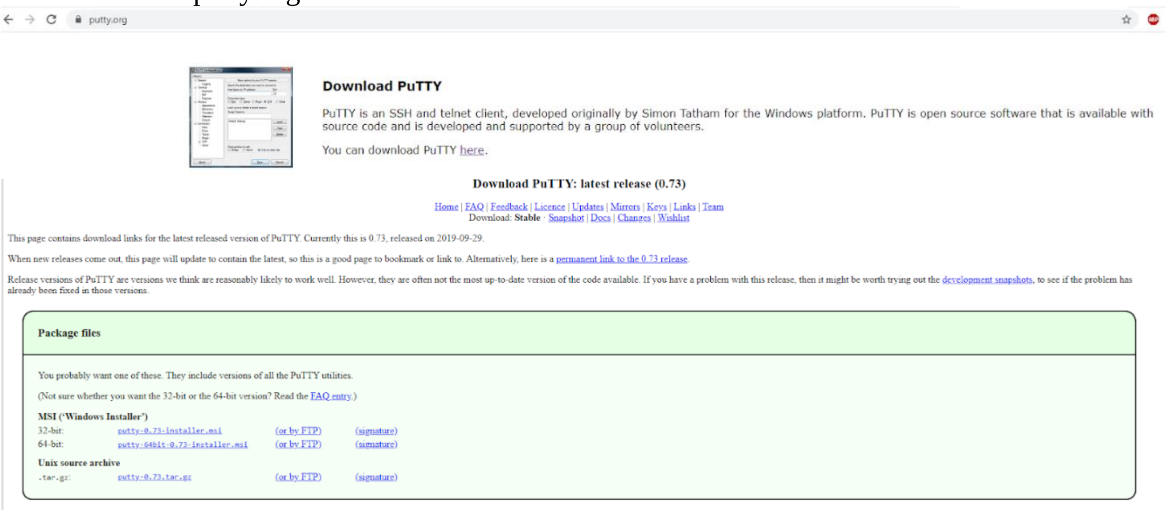
```
SQL> select * from fee_tot order by 1 fetch first 10 rows only;

FEE_TYPE        TOTAL
---------- ----------
Booking Ch      27275
arge

Group Book       1900
ing

No Charge           0
Research          500
```

Appendix H – Other Documentation

Remote Login Setup (SSH) - PuTTY & XMing

1. Download in the host machine PuTTY.

   putty.org



putty.org

Download PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers.

You can download PuTTY here.

Download PuTTY: latest release (0.73)

Home | FAQ | Feedback | Licence | Updates | Mirrors | Keys | Links | Team
Download: **Stable** · Snapshot | Docs | Changes | Wishlist

This page contains download links for the latest released version of PuTTY. Currently this is 0.73, released on 2019-09-29.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a permanent link to the 0.73 release.

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the development snapshots, to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include versions of all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry.)

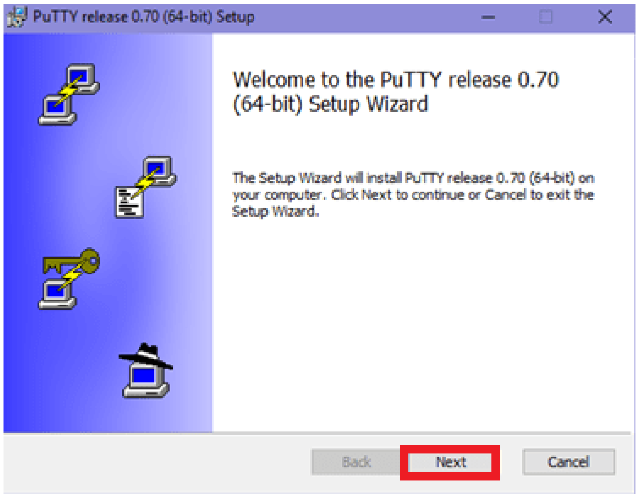
MSI ('Windows Installer')

32-bit:	putty-0.73-installer.msi	(or by FTP)	(signature)
64-bit:	putty-64bit-0.73-installer.msi	(or by FTP)	(signature)

Unix source archive

.tar.gz:	putty-0.73.tar.gz	(or by FTP)	(signature)

2. Start up the PuTTY installer. And click next.

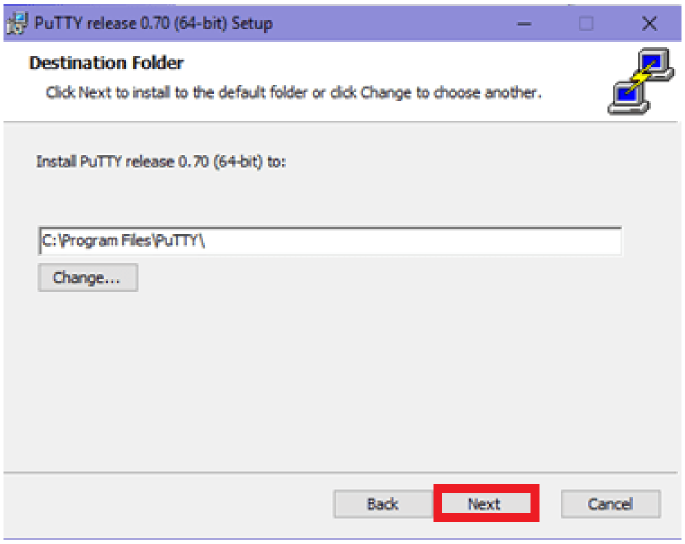


PuTTY release 0.70 (64-bit) Setup

Welcome to the PuTTY release 0.70 (64-bit) Setup Wizard

The Setup Wizard will install PuTTY release 0.70 (64-bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

3. Put in the desired location. Next.



PuTTY release 0.70 (64-bit) Setup

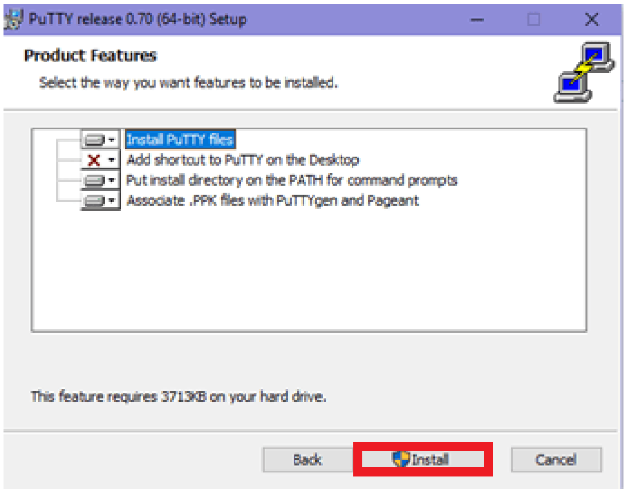
Destination Folder

Click Next to install to the default folder or click Change to choose another.

Install PuTTY release 0.70 (64-bit) to:

C:\Program Files\PuTTY\

4. Next.



PuTTY release 0.70 (64-bit) Setup

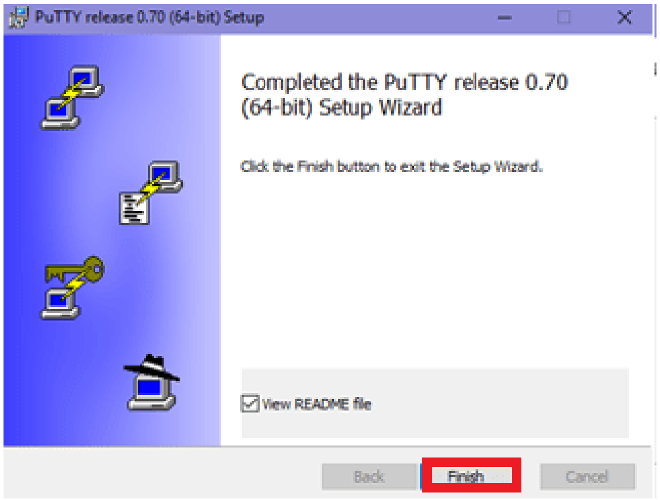
Product Features

Select the way you want features to be installed.

- Install PuTTY files
- Add shortcut to PuTTY on the Desktop
- Put install directory on the PATH for command prompts
- Associate .PPK files with PuTTYgen and Pageant

This feature requires 3713KB on your hard drive.

5. Finish.



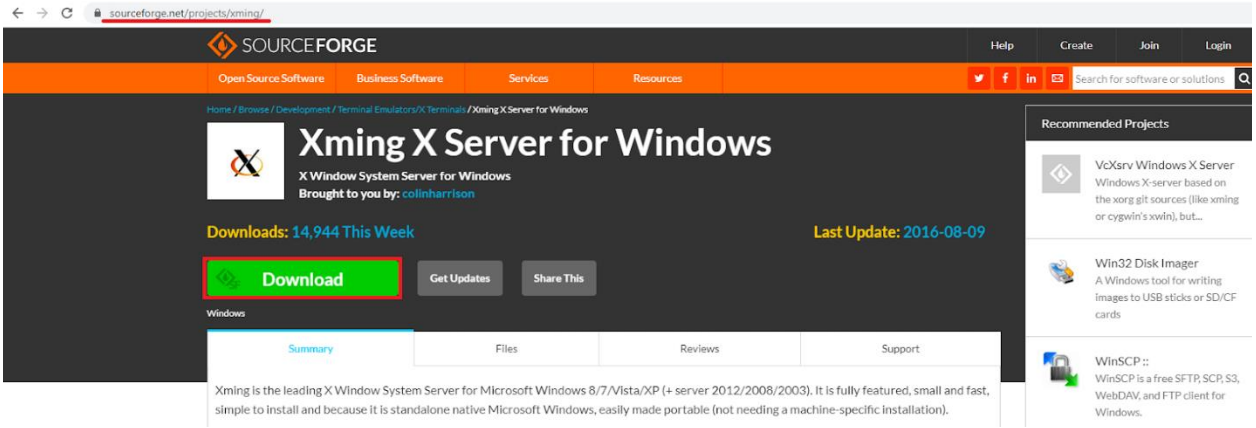
PuTTY release 0.70 (64-bit) Setup

Completed the PuTTY release 0.70 (64-bit) Setup Wizard

Click the Finish button to exit the Setup Wizard.

☑ View README file

6. Download the Xming installer from the website on the host.
   sourceforge.net/projects/xming



Home / Browse / Development / Terminal Emulators/X Terminals / Xming X Server for Windows

Xming X Server for Windows

X Window System Server for Windows

Brought to you by: colinharrison

Downloads: 14,944 This Week

Last Update: 2016-08-09

Xming is the leading X Window System Server for Microsoft Windows 8/7/Vista/XP (+ server 2012/2008/2003). It is fully featured, small and fast, simple to install and because it is standalone native Microsoft Windows, easily made portable (not needing a machine-specific installation).

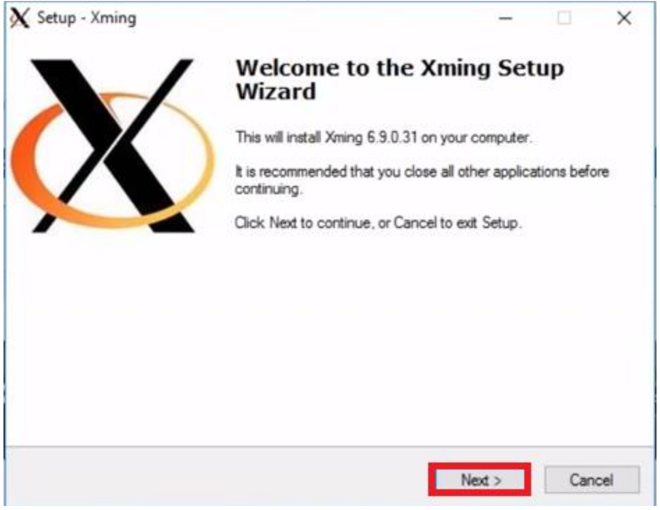
Recommended Projects

VcXsrv Windows X Server
Windows X-server based on the xorg git sources (like xming or cygwin's xwin), but...

Win32 Disk Imager
A Windows tool for writing images to USB sticks or SD/CF cards

WinSCP ::
WinSCP is a free SFTP, SCP, S3, WebDAV, and FTP client for Windows.

7. Startup the Xming installer. Click Next.



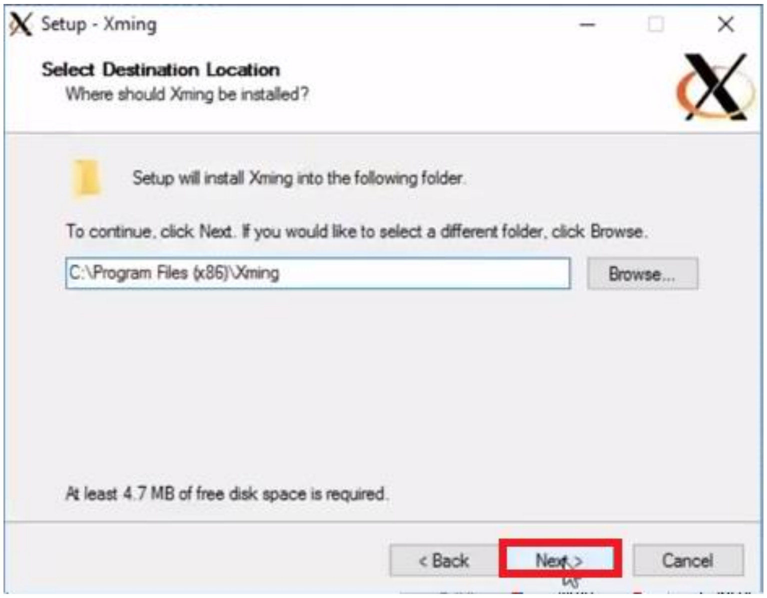
Welcome to the Xming Setup Wizard

This will install Xming 6.9.0.31 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

8. Enter the desired location for installation. Next.



Select Destination Location

Where should Xming be installed?

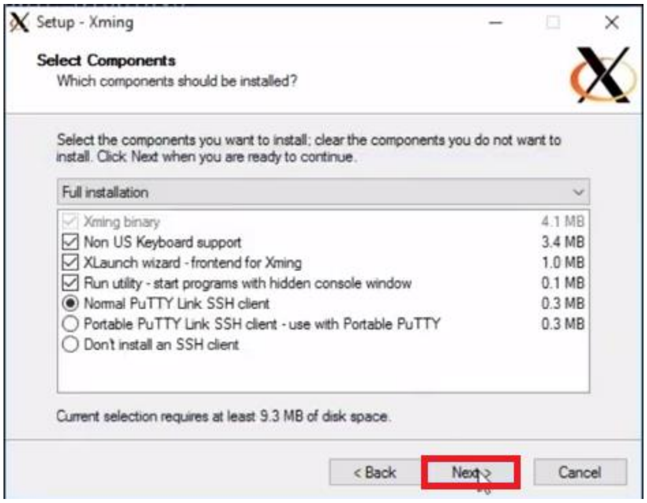
Setup will install Xming into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files (x86)\Xming

At least 4.7 MB of free disk space is required.

9. Next.


Setup - Xming

Select Components

Which components should be installed?

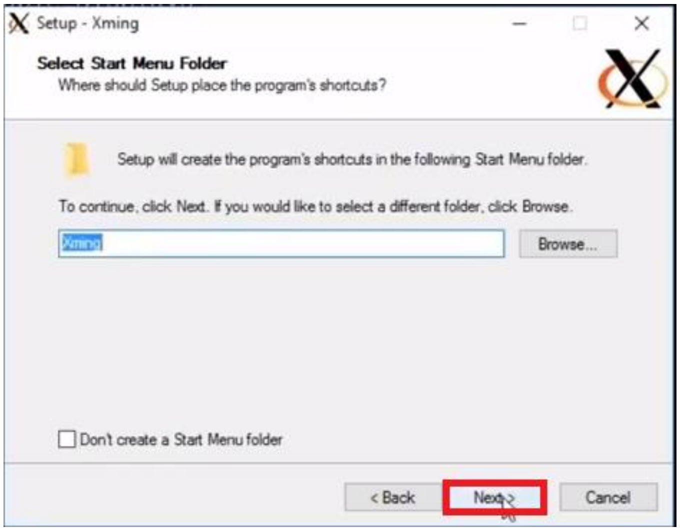
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Full installation

☑ Xming binary 4.1 MB
☑ Non US Keyboard support 3.4 MB
☑ XLaunch wizard - frontend for Xming 1.0 MB
☑ Run utility - start programs with hidden console window 0.1 MB
● Normal PuTTY Link SSH client 0.3 MB
○ Portable PuTTY Link SSH client - use with Portable PuTTY 0.3 MB
○ Don't install an SSH client

Current selection requires at least 9.3 MB of disk space.

10. Next.


Setup - Xming

Select Start Menu Folder

Where should Setup place the program's shortcuts?

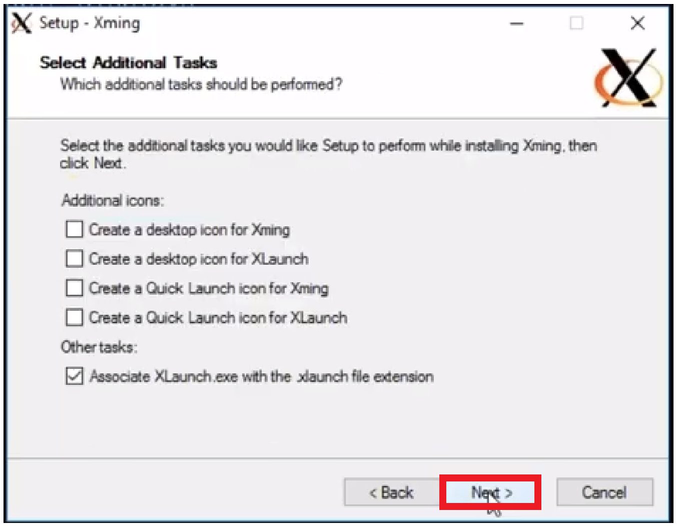
Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Xming

□ Don't create a Start Menu folder

11. Next.


Setup - Xming

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Xming, then click Next.

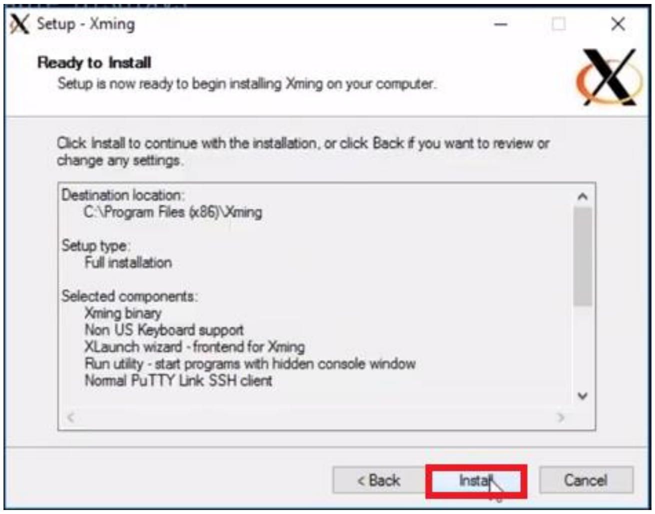
Additional icons:

□ Create a desktop icon for Xming
□ Create a desktop icon for XLaunch
□ Create a Quick Launch icon for Xming
□ Create a Quick Launch icon for XLaunch

Other tasks:

☑ Associate XLaunch.exe with the .xlaunch file extension

12. Install.



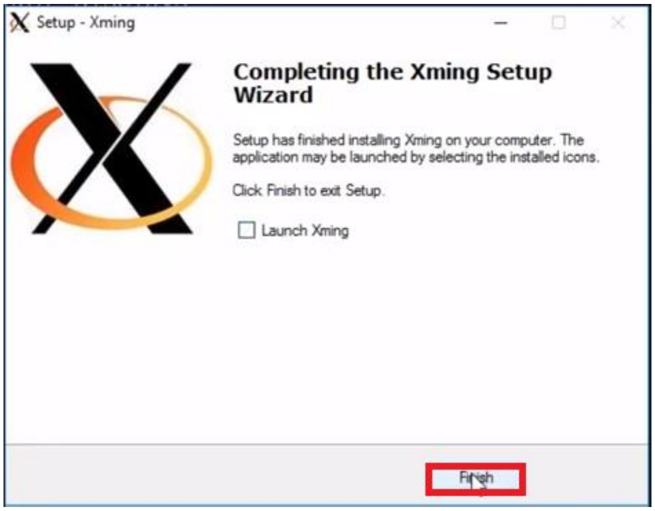
Ready to Install

Setup is now ready to begin installing Xming on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
    C:\Program Files (x86)\Xming

Setup type:
    Full installation

Selected components:
    Xming binary
    Non US Keyboard support
    XLaunch wizard - frontend for Xming
    Run utility - start programs with hidden console window
    Normal PuTTY Link SSH client
```

13. Finish.



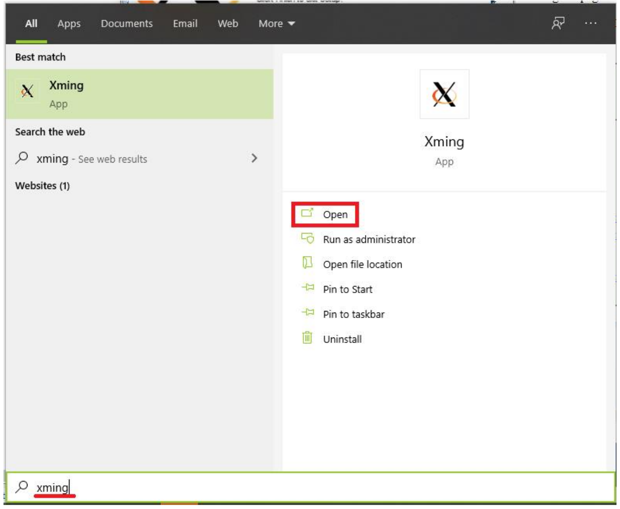
Completing the Xming Setup Wizard

Setup has finished installing Xming on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

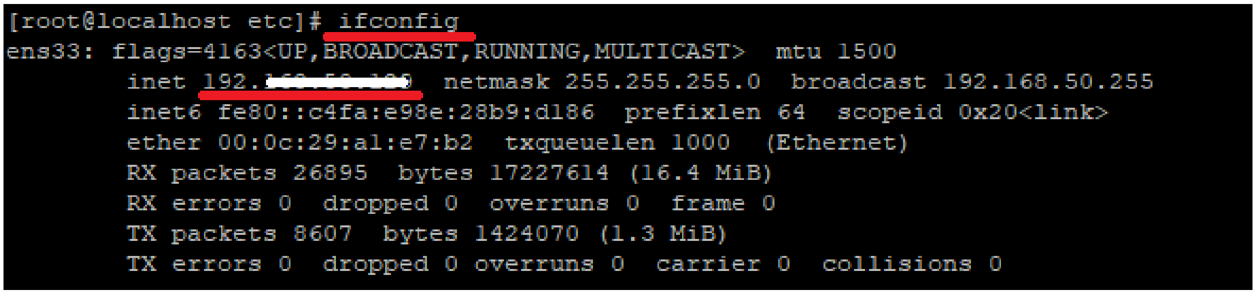
□ Launch Xming

14. Start up Xming



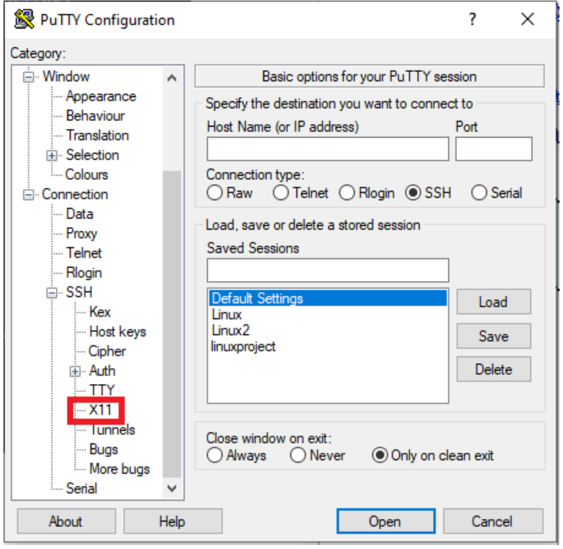
xming

15. Find the IP address of your server.



```
[root@localhost etc]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.[redacted]  netmask 255.255.255.0  broadcast 192.168.50.255
        inet6 fe80::c4fa:e98e:28b9:d186  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:a1:e7:b2  txqueuelen 1000  (Ethernet)
        RX packets 26895  bytes 17227614 (16.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8607  bytes 1424070 (1.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

16. Start up putty. Choose the X11 under the Category



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) Port

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

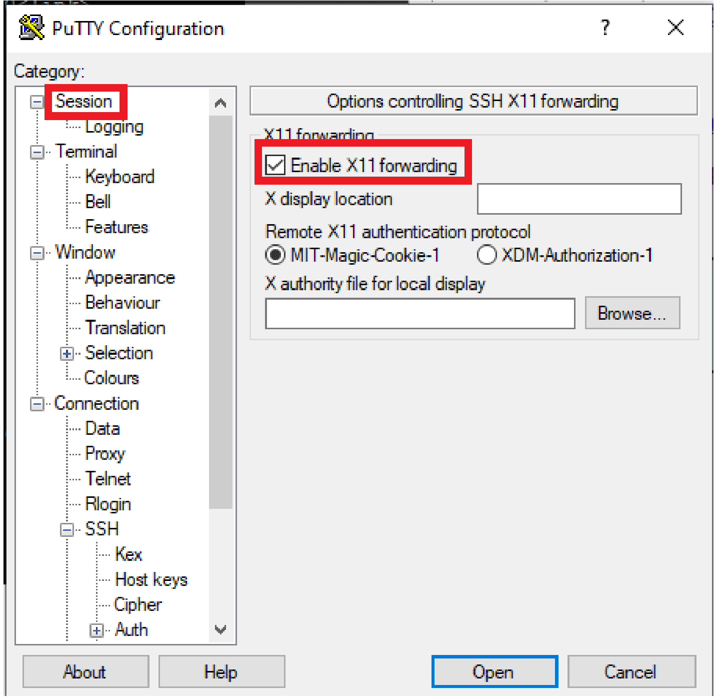
Load, save or delete a stored session

Saved Sessions

Default Settings
Linux
Linux2
linuxproject

Close window on exit: ○ Always ○ Never ● Only on clean exit

17. Enable the X11 forwarding. Go back to the session.



Options controlling SSH X11 forwarding

X11 forwarding

☑ Enable X11 forwarding

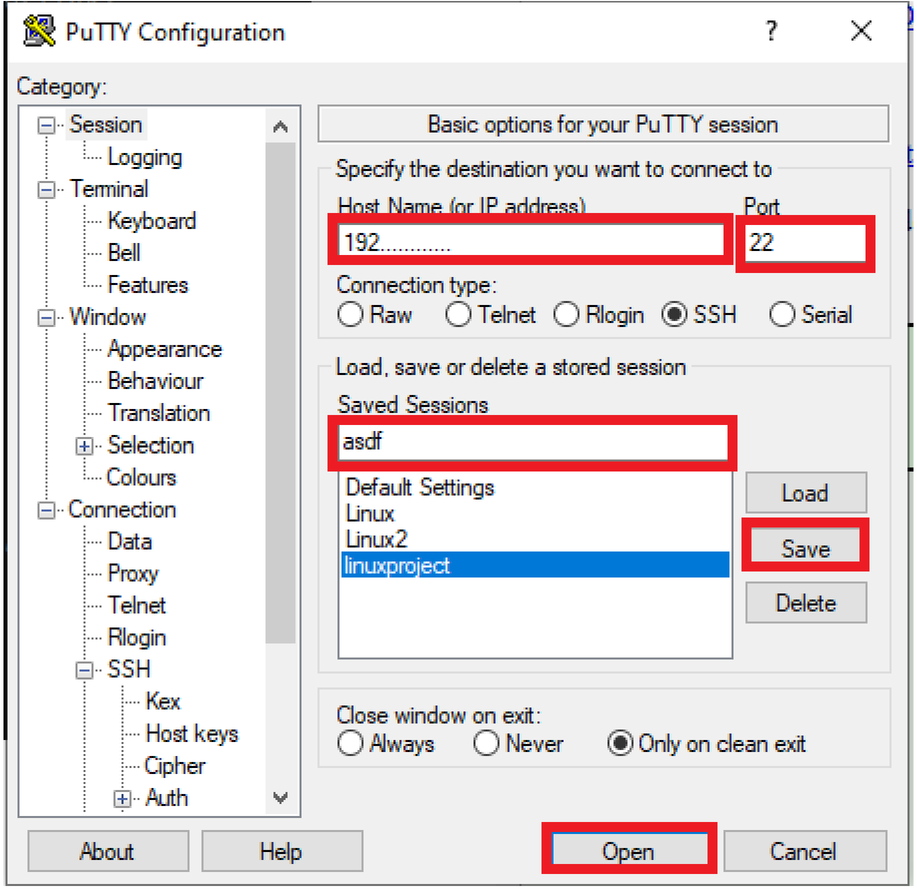
X display location

Remote X11 authentication protocol
● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

X authority file for local display

18. Enter the IP address from the server
19. Enter the port as 22 for SSH connection.
20. Enter a name for the session for future ease of access.

21. Save the session.
22. Open the session.



Specify the destination you want to connect to

Host Name (or IP address): 192...........

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

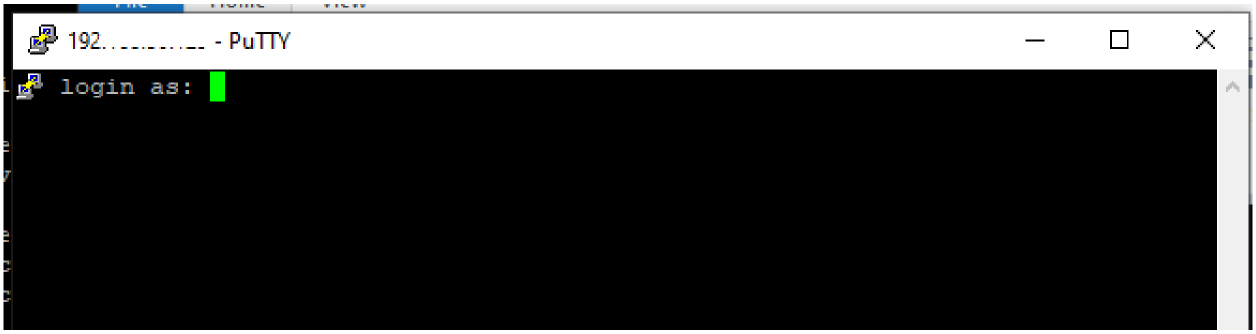
Load, save or delete a stored session

Saved Sessions: asdf

Default Settings
Linux
Linux2
linuxproject

Close window on exit: ○ Always ○ Never ● Only on clean exit

23. Enter a username to log in as. Must have the server running, with network available and connected on both host and server.



```
login as:
```