SQL Lab

- Setting up Oracle Chinook

In this section you will begin the process of working with the Oracle Chinook database
Task – Open the Chinook_Oracle.sql file and execute the scripts within.

2.0 SQL Queries

In this section you will be performing various queries against the Oracle Chinook database.

2.1 SELECT

Task – Select all records from the Employee table.

```
Select * from EMPLOYEE;
```

Task – Select all records from the Employee table where last name is King.

```
select * from EMPLOYEE WHERE LASTNAME = ‘King’;
```

Task – Select all records from the Employee table where first name is Andrew and REPORTSTO is NULL.

```
select * from EMPLOYEE WHERE FIRSTNAME = 'Andrew' AND REPORTSTO IS NULL;
```

2.2 ORDER BY

Task – Select all albums in Album table and sort result set in descending order by title.

```
Select * from ALBUM ORDER by TITLE DESC;
```

Task – Select first name from Customer and sort result set in ascending order by city

```
Select * from CUSTOMER ORDER BY CITY ASC;
```

2.3 INSERT INTO

Task – Insert two new records into Genre table

```
INSERT INTO GENRE VALUES(26,'Smooth jazzy jazz');
```

Task – Insert two new records into Employee table

```
INSERT INTO EMPLOYEE
VALUES(9,'Smitherman','Austin','Boss',NULL,TO_DATE('1968-01-09','YYYY-MM-
DD'),TO_DATE('2004-03-04', 'YYYY-MM-DD'),'923 7 ST NW','Austin Town','CA','USA','T1H
1Y8','+1 (403) 467-3351','+1 (403) 467-8772','laura@chinookcorp.com');
INSERT INTO EMPLOYEE
VALUES(10,’Smitherman','Austin’,’Janitor’,NULL,TO_DATE('1968-01-09','YYYY-MM-
DD'),TO_DATE('2004-03-04', 'YYYY-MM-DD'),'923 7 ST NW','Austin Town','CA','USA','T1H
1Y8','+1 (403) 467-3351','+1 (403) 467-8111’,’Austin@chinookcorp.com');
```

Task – Insert two new records into Customer table

```
INSERT INTO CUSTOMER VALUES(60,'Austin','Smithermon','Austin''s biz','923 7 ST
NW','Austin Town','CA','USA','T1H 1Y8','+1 (403) 467-3351','+1 (403)
467-8772','laura@chinookcorp.com',4);
INSERT INTO CUSTOMER VALUES(61,’Austin','SmitHermAn','Austin''s biz','923 7 ST
NW','Austin Town','CA','USA','T1H 1Y8','+1 (403) 467-3351','+1 (403)
467-8772','laura@chinookcorp.com',4);
```

2.4 UPDATE

Task – Update Aaron Mitchell in Customer table to Robert Walter

```
Update CUSTOMER SET FIRSTNAME = 'Robert', LASTNAME = 'Walter' Where
```

```
FIRSTNAME = 'Aaron' AND LASTNAME = 'Mitchell';
```

Task – Update name of artist in the Artist table “Creedence Clearwater Revival” to “CCR”

```
UPDATE ARTIST SET NAME = 'CCR' WHERE NAME = 'Creedence Clearwater Revival';
```

2.5 LIKE

Task – Select all invoices with a billing address like “T%”

```
SELECT * from INVOICE Where BILLINGADDRESS LIKE 'T%';
```

2.6 BETWEEN

Task – Select all invoices that have a total between 15 and 50

```
SELECT * from INVOICE Where TOTAL > 15 AND TOTAL < 50;
```

Task – Select all employees hired between 1st of June 2003 and 1st of March 2004

```
SELECT * from EMPLOYEE where HIREDATE between TO_DATE('01-06-2003', 'dd-mm-yyyy') AND TO_DATE('01-03-2004', 'dd-mm-yyyy');
```

2.7 DELETE

Task – Delete a record in Customer table where the name is Robert Walter (There may be constraints that rely on this, find out how to resolve them).

```
alter table INVOICE
drop constraint FK_INVOICECUSTOMERID;

alter table INVOICE
add constraint FK_INVOICECUSTOMERID
foreign key (CUSTOMERID)
references CUSTOMER (CUSTOMERID)
on delete cascade;

alter table INVOICELINE
drop constraint FK_INVOICELINEINVOICEID;

alter table INVOICELINE
add constraint FK_INVOICELINEINVOICEID
foreign key (INVOICEID)
references INVOICE (INVOICEID)
on delete cascade;

DELETE FROM CUSTOMER WHERE FIRSTNAME = 'Robert' AND LASTNAME = 'Walter';
```

7.0 JOINS

In this section you will be working with combing various tables through the use of joins. You will work with outer, inner, right, left, cross, and self joins.

7.1 INNER

Task – Create an inner join that joins customers and orders and specifies the name of the customer and the invoiceId.

```
SELECT cust.FIRSTNAME, cust.LASTNAME, inv.INVOICEID
```

```
FROM (SELECT FIRSTNAME, LASTNAME, CUSTOMERID FROM CUSTOMER) cust
INNER JOIN (SELECT INVOICEID,CUSTOMERID FROM INVOICE) inv
on cust.CUSTOMERID = inv.CUSTOMERID;
```

7.2 OUTER

Task – Create an outer join that joins the customer and invoice table, specifying the CustomerId, firstname, lastname, invoiceId, and total.

```
SELECT cust.CUSTOMERID, cust.FIRSTNAME, cust.LASTNAME, inv.INVOICEID,
inv.TOTAL
FROM (SELECT FIRSTNAME, LASTNAME, CUSTOMERID FROM CUSTOMER) cust
FULL OUTER JOIN (SELECT INVOICEID,CUSTOMERID,TOTAL FROM INVOICE) inv
on cust.CUSTOMERID = inv.CUSTOMERID;
```

7.3 RIGHT

Task – Create a right join that joins album and artist specifying artist name and title.

```
SELECT art.NAME, alb.TITLE
FROM (SELECT * FROM ARTIST) art
RIGHT JOIN (SELECT * FROM ALBUM) alb
on art.ARTISTID = alb.ARTISTID;
```

7.4 CROSS

Task – Create a cross join that joins album and artist and sorts by artist name in ascending order.

```
SELECT * FROM ARTIST CROSS JOIN ALBUM ORDER BY ARTIST.NAME ASC;
```

7.5 SELF

Task – Perform a self-join on the employee table, joining on the reportsto column.

```
SELECT * FROM EMPLOYEE emp, EMPLOYEE man
        WHERE emp.REPORTSTO = man.EMPLOYEEID;
```