# Proposed of Project

For my project proposal, I created a database system about plane ticket reservation. This database system will show the records of the customers who have reserved their ticket via phone. There will be three records, customer records which contains the customer’s id, name and email address. Employees’ records, this record contains about the employees information. And lastly, the tickets’ records, this will records display the ticket id, customer’s id, their origin and destination and the departure and return date.

# Database Description

EMP\_TBL

Emp\_id [pk]

Emp\_fname

Emp\_lname

Emp\_bonus

Emp\_sal

Emp\_netsal

TCKT\_TBL

Tckt\_num [pk]

Cust\_id [fk] ]

Emp\_id [fk]

Departure

Return

Destination

Origin

CUST\_TBL

Cust\_id [pk] ]

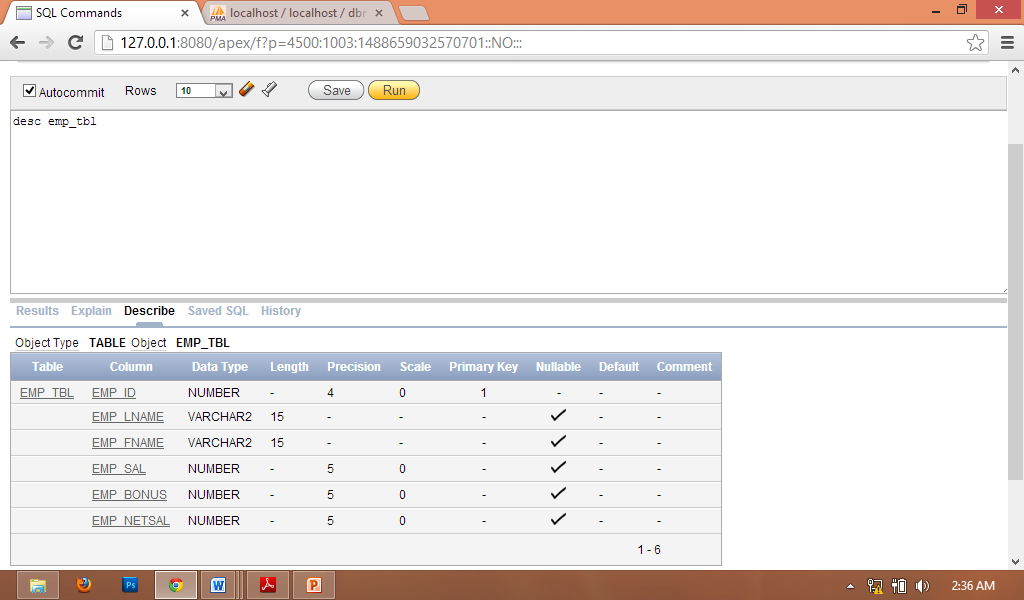
Cust\_fname

Cust\_lname

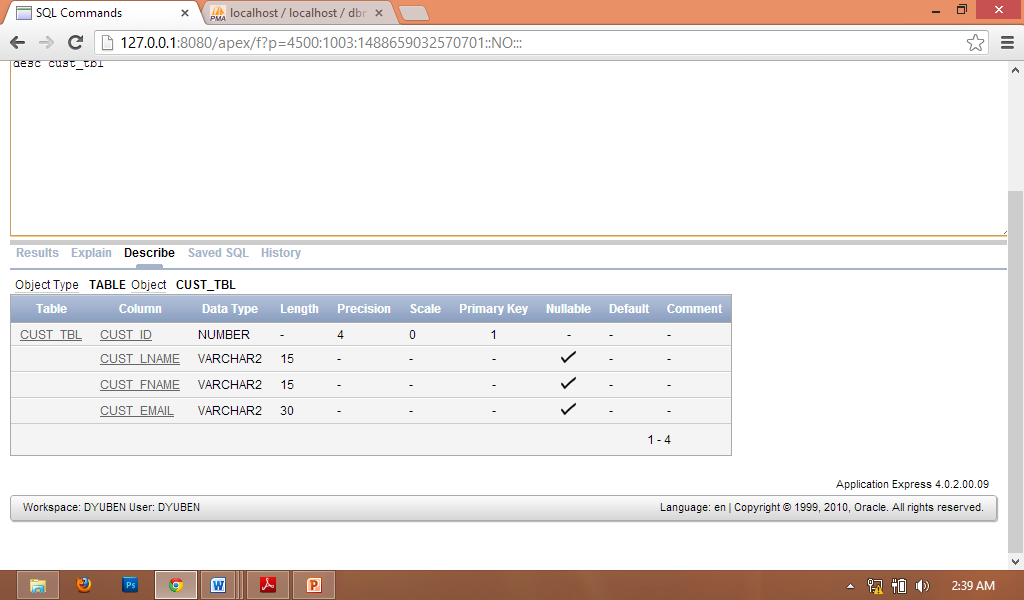
Cust\_email [uk]

# Table Description

* Employee’s table is consists of employee’s information. Id number, employee’s name with their salary, bonus and net salary. Type in the script “desc <table name>” to display table description. I set the column **emp\_id**  as the primary key and as a null value.



* Customer’s table is basically consists of customers information. Customer id, customers name and email address. Type in the script “desc <table name>” to display table description. I set the column **cust\_id** as a primary key and as a null value. I set the **cust\_email** as a unique key.



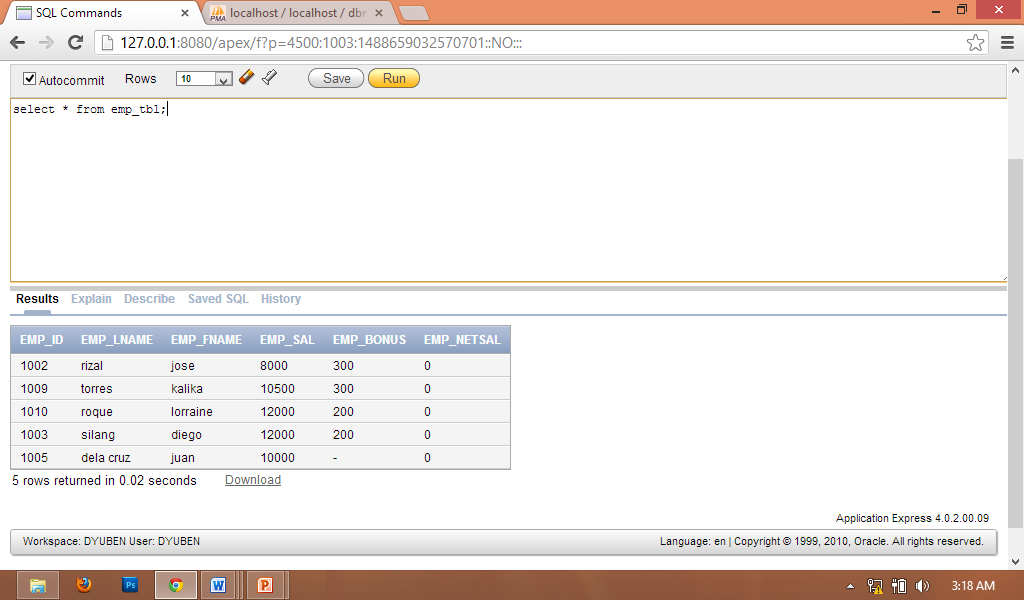
* Ticket’s table are consists of ticket number, customers id for customers information, employee’s id to locate the employee who inserted the customers reservation, origin and destination for the place where they will head to, and lastly departure and return for the date when they will depart and arrive. I set **tckt\_num**  as its primary key and has a null value. **emp\_id** and **cust\_id** has a null value also.



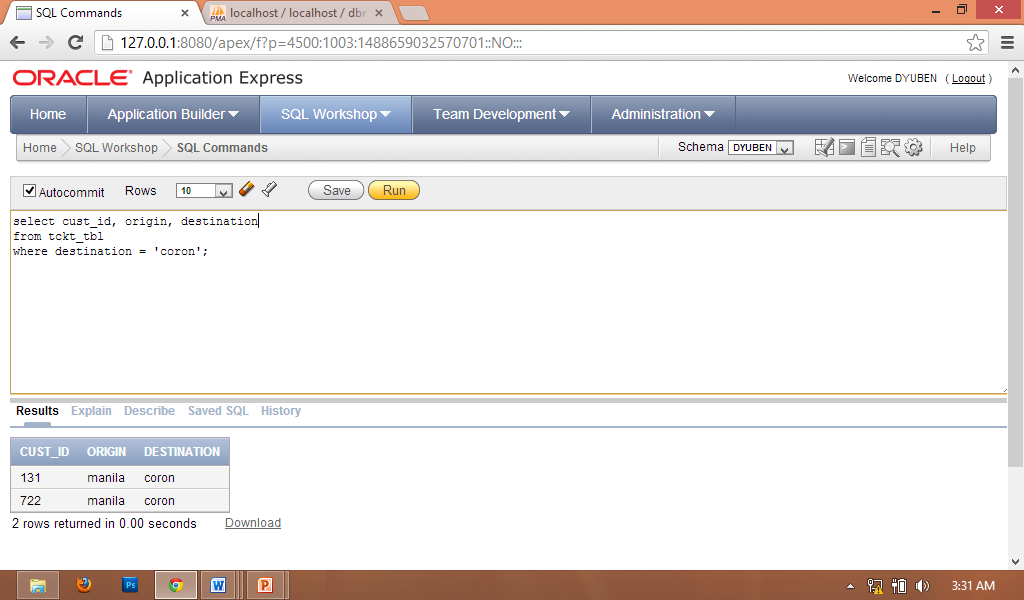
# SQL Command Listing

**Restriction and Sorting**

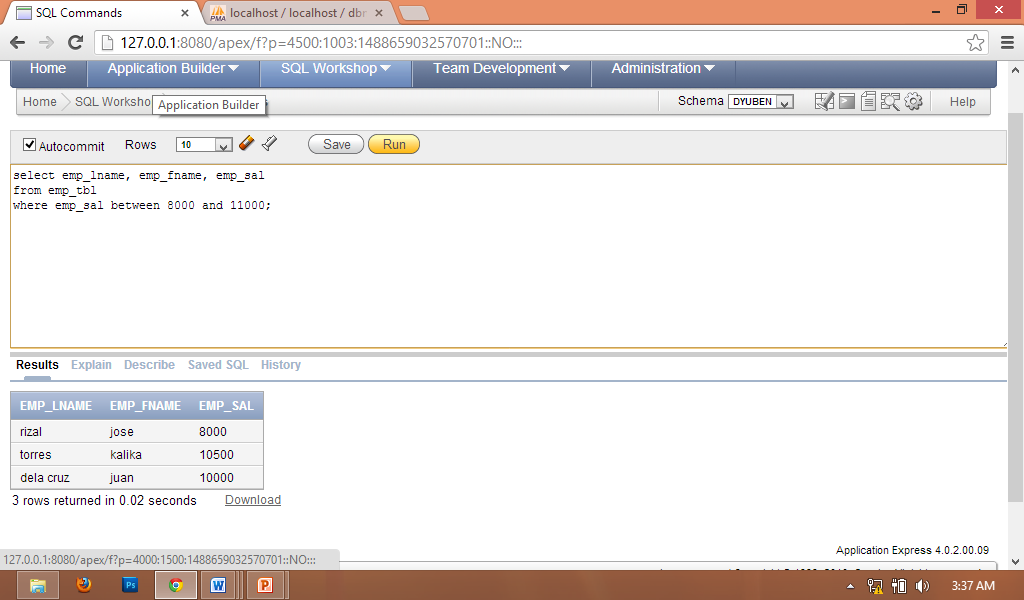
* **Select Statement** enables you to display the records from your table. Select identifies what columns while from identifies which table. “\*” is use for displaying all the records in the table.



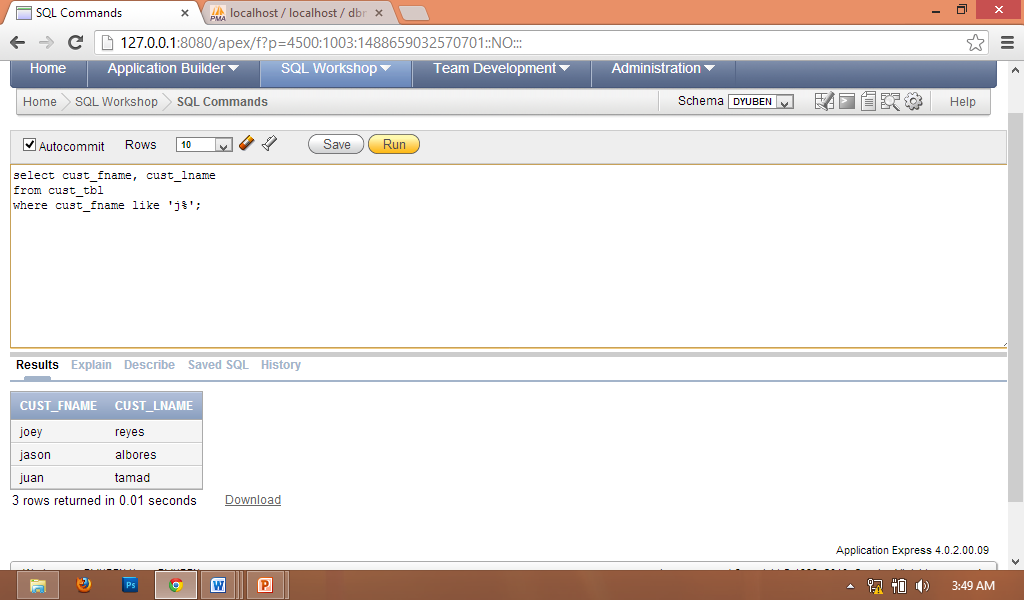
* **Where Clause** enables you to limit the query to rows that meet a condition. While condition is composed of column names, expressions and comparison operators.



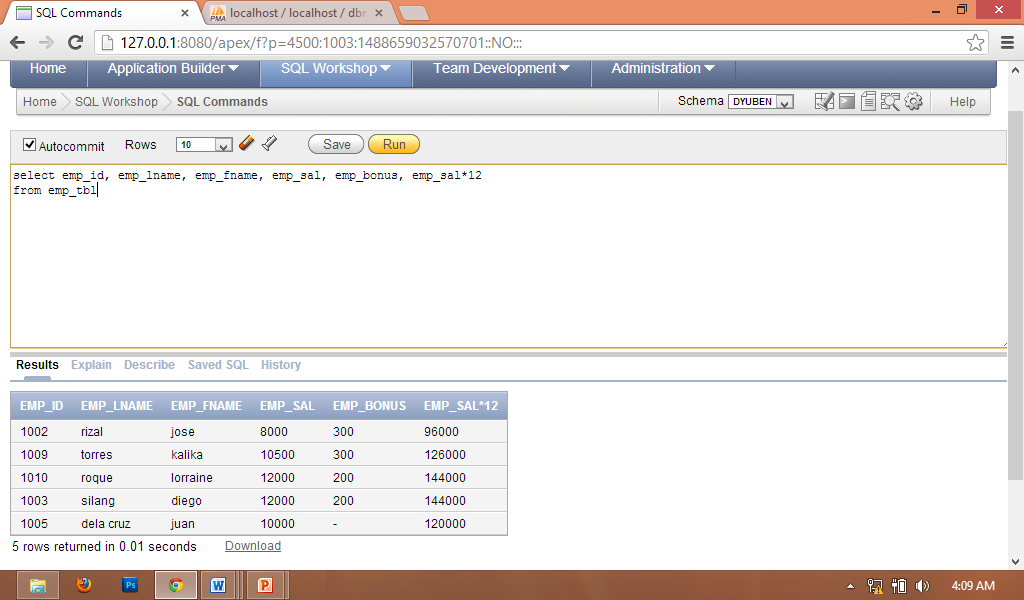
* **Between Clause** displaying rows based on a range of values. Contains a lower limit and an upper limit. Values specified with the **between** condition are inclusive.



* **Like function** perform wildcard searches of valid search string values.

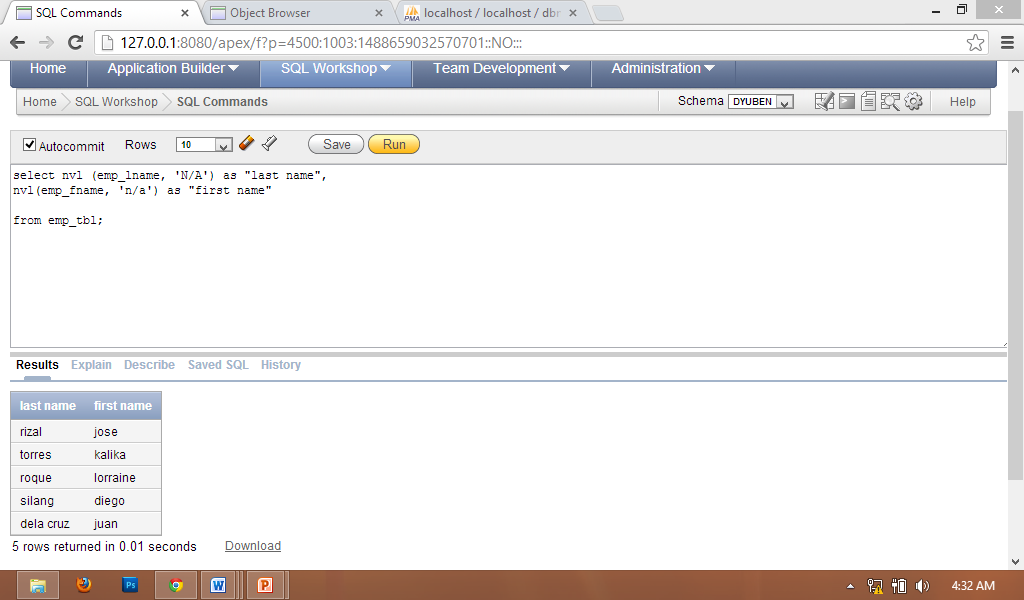


* **Arithmetic Expressions and Operation** enable us to use arithmetic operators such as “+, - , \*, /”.

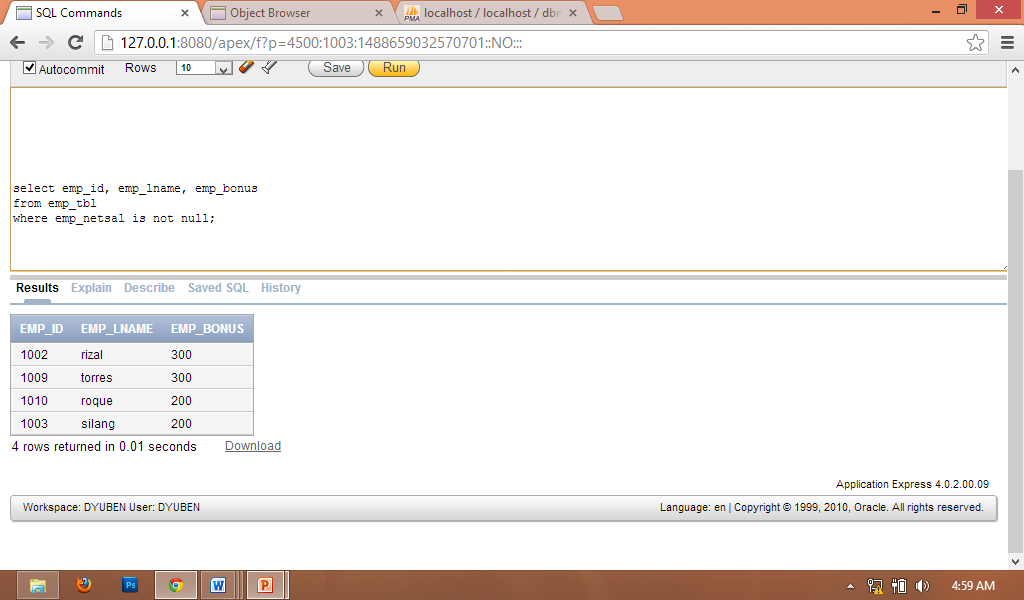
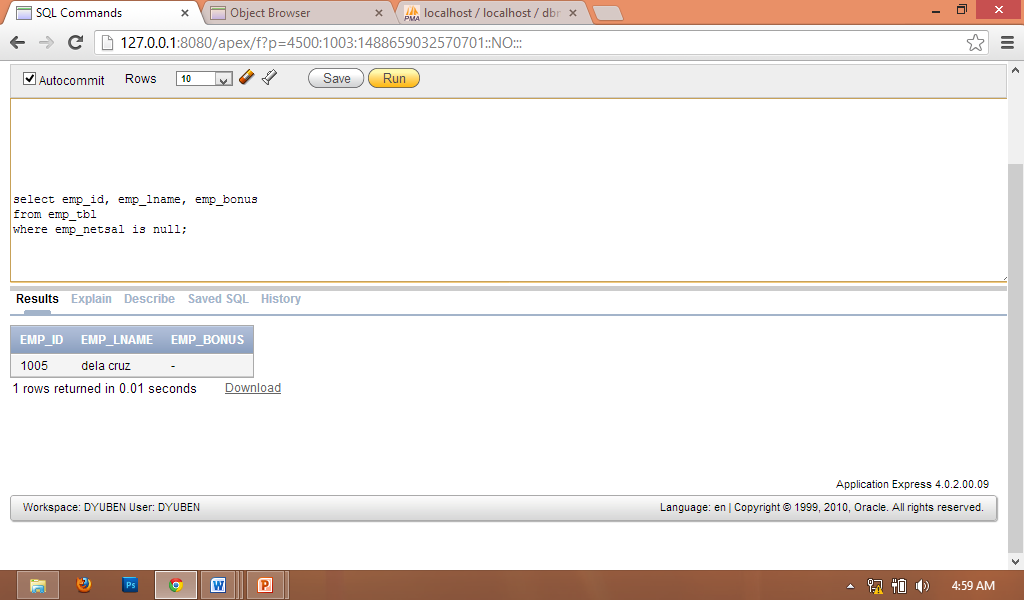


# Null Value Handling

* **NVL functions** use to specific display the rows with a null value.



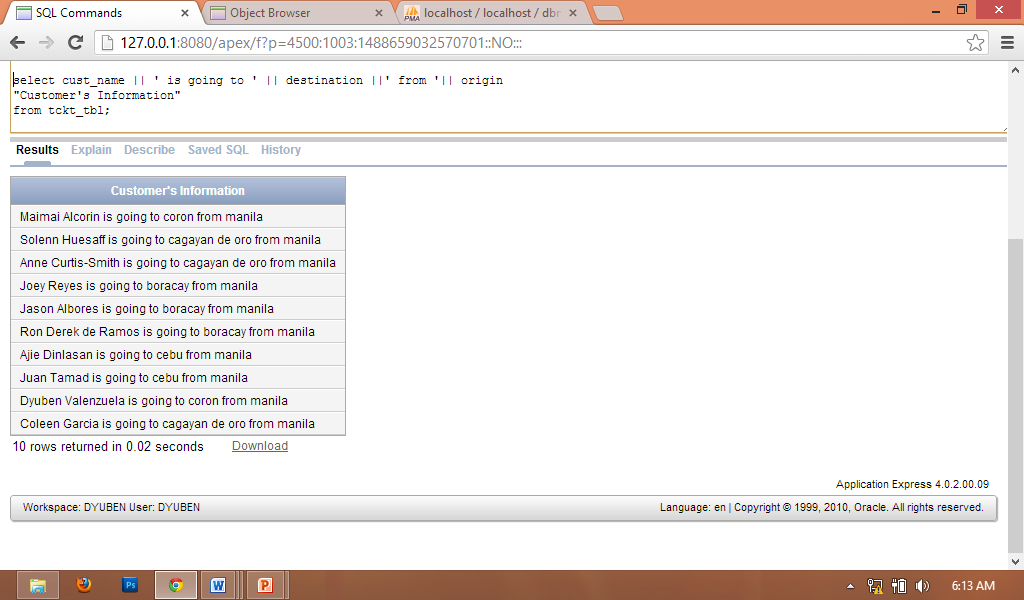
* **IS NOT NULL/ IS NULL VALUE** uses to display the rows with a data type as not null or a data type as a null value.

**Aliases** use to rename column heading.

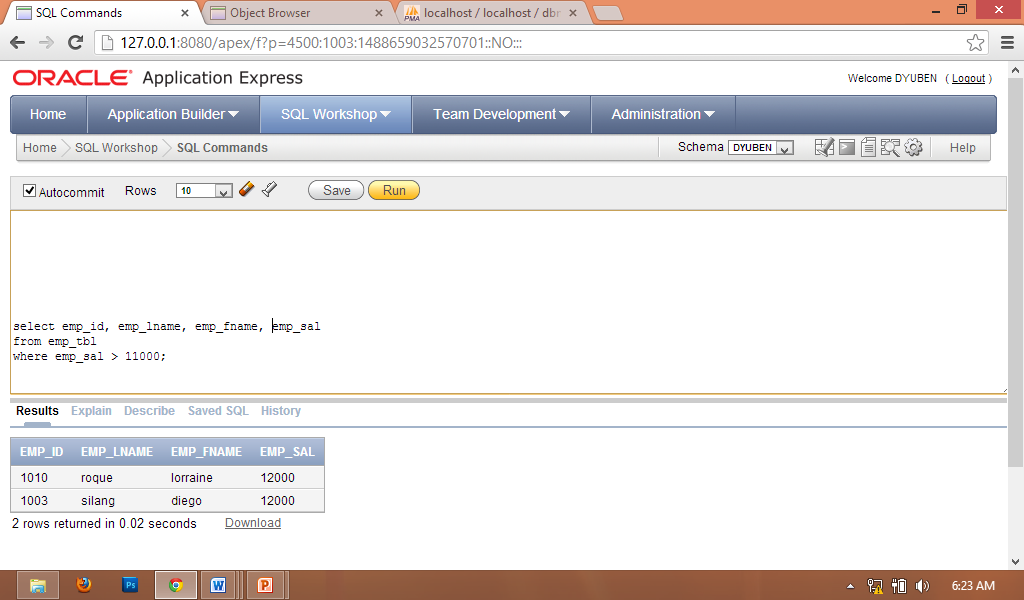


**Concatenations** use for combining columns to other columns.



**Comparison Operators**

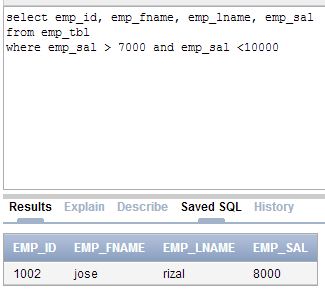
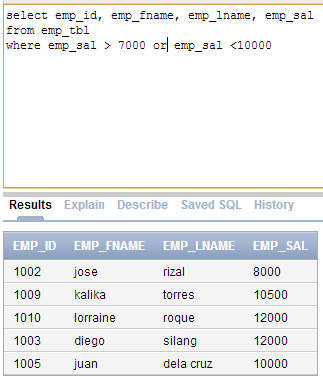
|  |  |
| --- | --- |
| OPERATOR | DESCRIPTION |
| > | Greater than |
| < | Less than |
| >= | Greater than or equal |
| <= | Less than or equal |
| = | Is equal to |
| != | Is not equal to |



**LOGICAL OPERATOR**

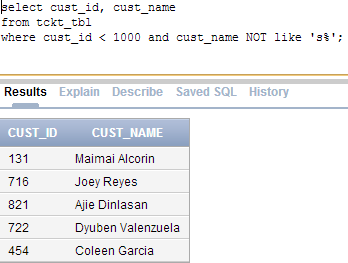
|  |  |
| --- | --- |
| OPERATOR | DESCRIPTION |
| AND | Greater than |
| OR | Less than |
| NOT | Greater than or equal |

**AND**  **OR**



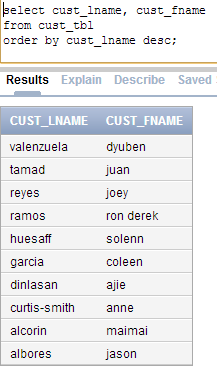
­

**NOT**



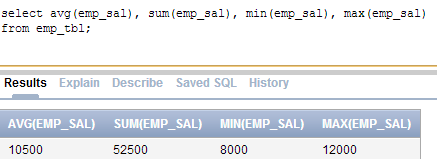
# Sorting

* **ORDER BY** clause is used to arrange the rows by ascending “ASC” and descending “DESC”.

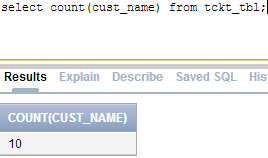


# Single row Function

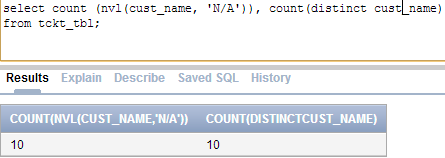
* **AVG** used to get the average of the specific columns
* **SUM** used to get total of the specific columns
* **MIN** used to display the minimum record among the columns
* **MAX** used to display the maximum record among the columns



* **COUNT** obviously it is used to count the records inside the column

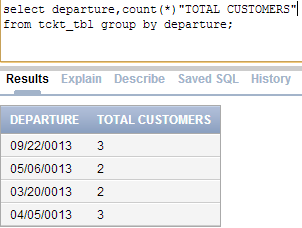


* **DISTINCT** used to count distinct non null values
* **COUNT w/ NVL** used to count even the null values



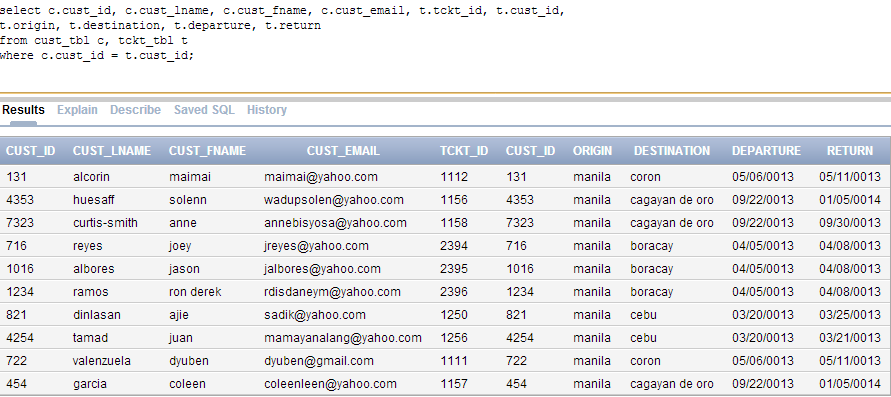
# Group Functions

* **GROUP BY** used to display the selected conditions by group

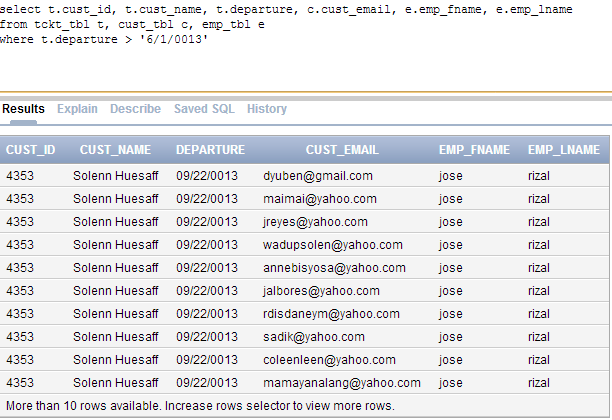


# Joining Tables

* **EQUIJOIN** used to join two tables. They must be related to each other. i.e. Primary key to Foreign key

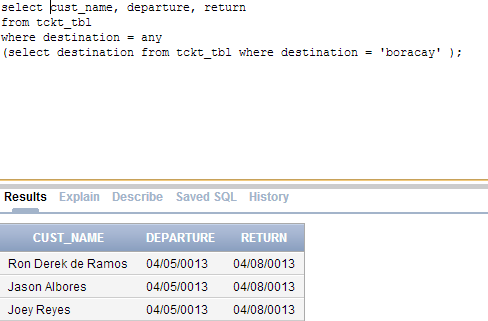


* **NON EQUIJOIN**



# Subqueries

The main subquery is equal to the inner subquery condition



# TESTING

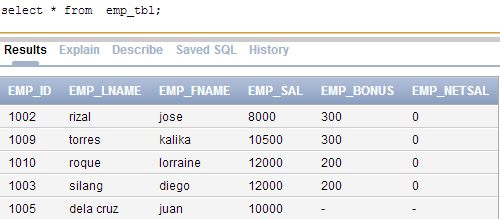
TEST PLAN

|  |  |
| --- | --- |
| Test Cases | Objective |
| 1 | To test if you can view all records on the employee table |
| 2 | To test if you can view all records on the ticket table |
| 3 | To test if you can view all records on the customer table |
| 4 | To test if you could view the employee table’s properties using DESC |
| 5 | To test if you could view the ticket table’s properties using DESC |
| 6 | To test if you could view the customer table’s properties using DESC |
| 7 | To test if you can view the details in descending order |

TEST CASE 1

Objective: To test if you can view all records on the employee table

Expected Result: After entering the syntax below, you can view the record on employee table



TEST CASE 2

Objective: To test if you can view all records on ticket table

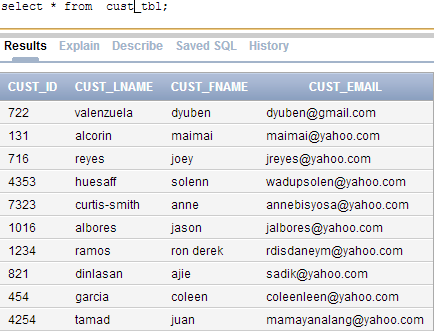
Expected Result: After entering the syntax below, you can view the record on ticket table



TEST CASE 3

Objective: To test if you can view all records on customer table

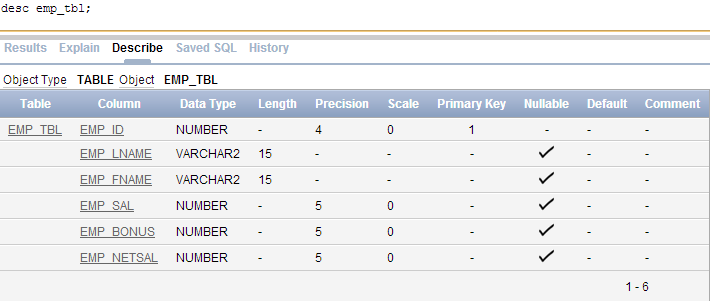
Expected Result: After entering the syntax below, you can view the record on customer table



TEST CASE 4

Objective: To test if you could view the employee table’s properties using DESC

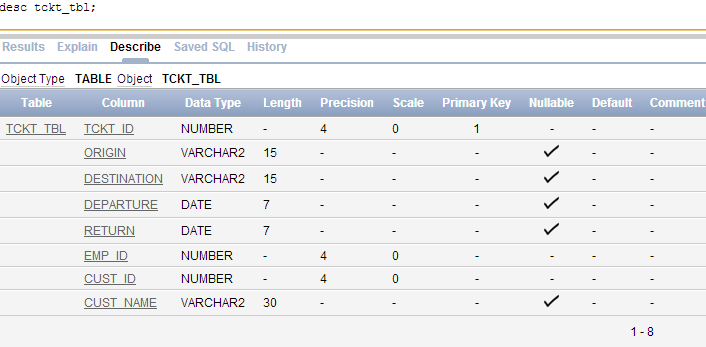
Expected Result: After entering the syntax below, you can view the properties of employee table



TEST CASE 5

Objective: To test if you could view the ticket table’s properties using DESC

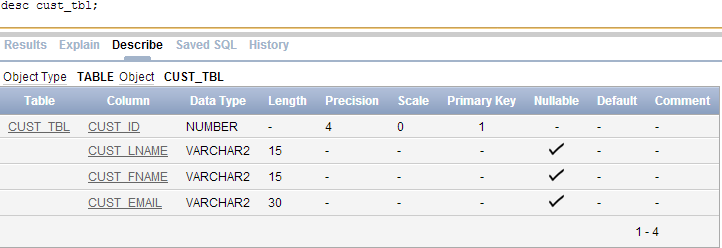
Expected Result: After entering the syntax below, you can view the properties of ticket table



TEST CASE 6

Objective: To test if you could view the customer table’s properties using DESC

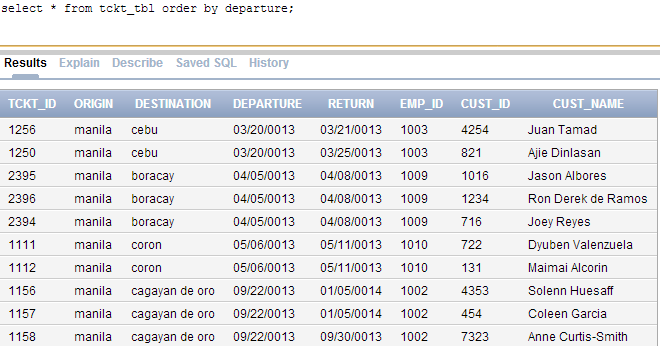
Expected Result: After entering the right codes, you can view the properties of customer table



TEST CASE 7

Objective: To test if you can view the details in descending order

Expected Result: After entering the syntax below, you can view the records in descending order



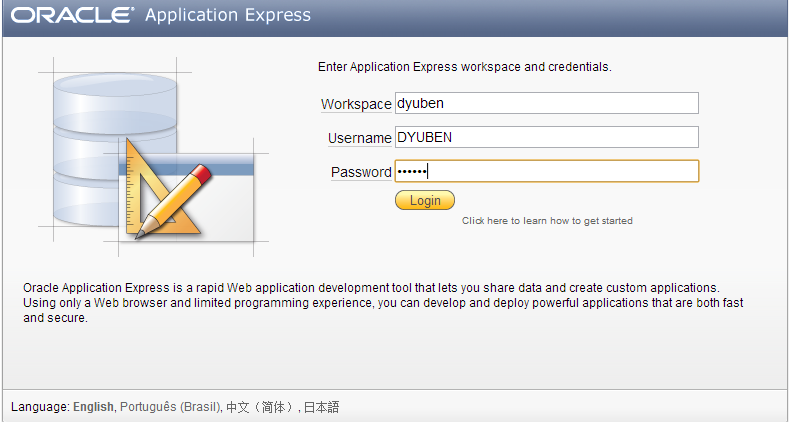
# TEST RESULTS

|  |  |
| --- | --- |
| Test Cases | Test Result |
| 1 | Successful |
| 2 | Successful |
| 3 | Successful |
| 4 | Successful |
| 5 | Successful |
| 6 | Successful |
| 7 | Successful |

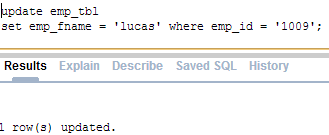
# USER MANUAL

The proponent use Oracle Application Express

Log in your username and password.



**To update the table**



**To insert a record To delete a record**

