

## **DATABASE DESIGN FOR AN APARTMENT RENTAL LISTINGS**

### **TEAM:**

Sunit Bail : 001577432, Rochak Nath :001563060, Ravinder Rao Madhavaram : 001541885,  
Nikhil Nethaji Reddy Mallepally : 001541881

### **OBJECTIVES:**

To Design a Database for Apartment Rental Services which supports following features:

- ❖ Users must register and login with details like Email, Username, Password, Phone Number.
- ❖ Users are allowed to upload their listings as well as view new and existing listings.
- ❖ Users can filter their priorities/preferences to find a suitable property.
- ❖ Users who are a premium account holders will get to know the upcoming deals before the rest of the users.
- ❖ Renting/Leasing details must be stored, Users and Employees must be able to access the details.
- ❖ Details of the Employees must be maintained such as email, firstname, lastname, designation, phone number etc.

### **PROBLEM STATEMENT:**

- 1) User may switch the account from normal to premium and vice-versa, which will be difficult to track manually .
- 2) Unable to retrieve all the listings proposed by a particular user, since users and listings are stored in separate files.
- 3) Multiple users are signing-up with the same username and it is difficult to prevent this by manually searching the existing usernames. This causes data inconsistency.
- 4) Difficult to find the deal proposed by a user for one of his/her listing details manually.

### **PROPOSED SOLUTIONS:**

1. Specific listings are marked as premium, which would mean that customers with a subscription ID are allowed access to those listings. At the end of the subscription period, their accounts would revert back automatically to normal status.
2. The database design will help to link the Users and Listings with the help of a FOREIGN KEY in this case which will be the POSTER ID key which will act as a bridge table and connect users to listings and a smooth data flow and consistency.
3. All the users will be provided with a CUSTOMER-ID, where ID being the primary key hence giving every user a unique number. The database clearly shows that in the customer table that as soon as you register you would be provided with a key.
4. In the design there is a separate table for the tags where the users can change the tags as well as rent according to their need and change to their convenience.

## BUSINESS RULES

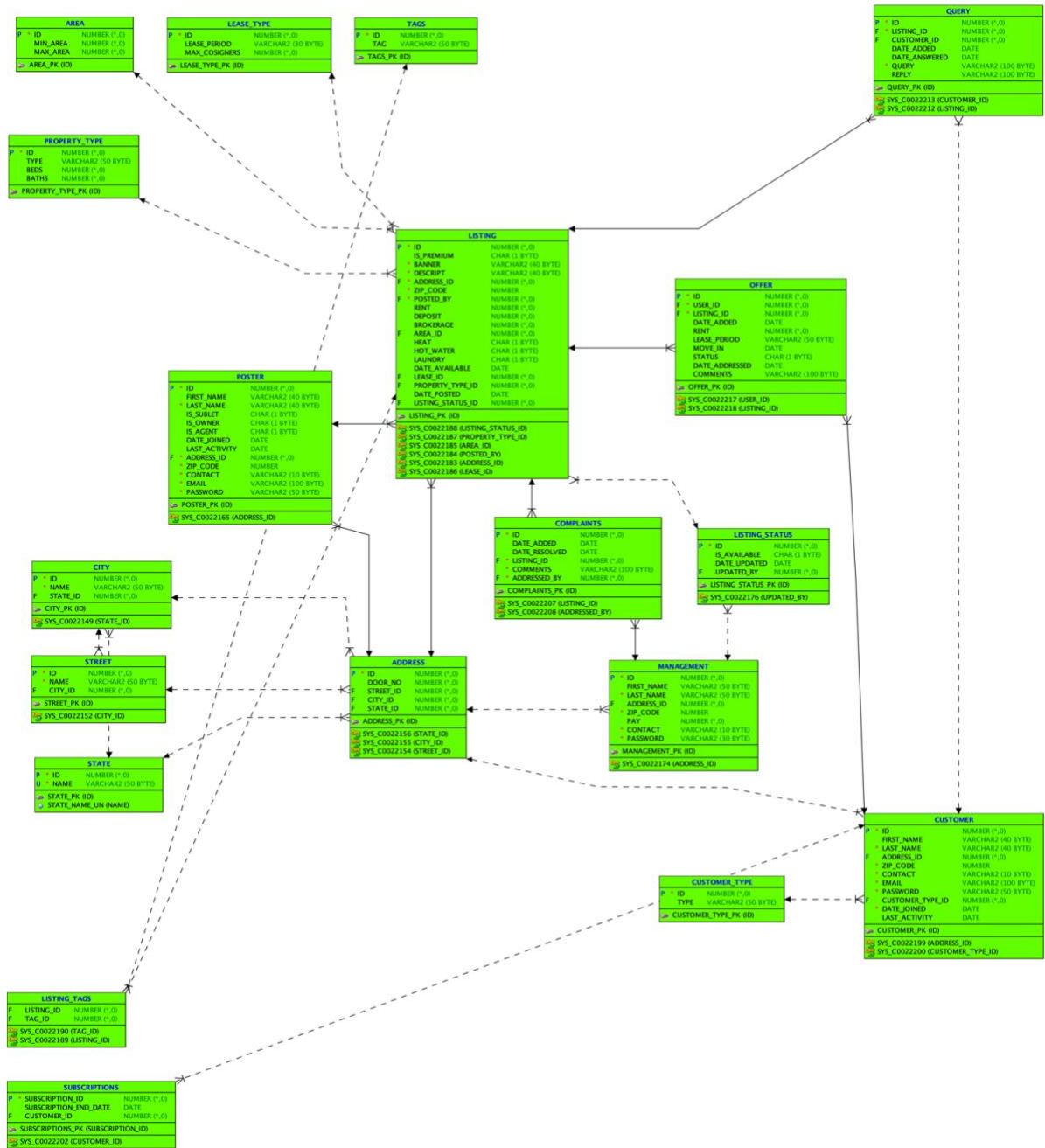
1. USER can create one or more LISTINGS.
2. A LISTING can be created by only one USER.
3. Every LISTING has its own ID associated with it.
4. LISTINGS may or may not have OFFERS it depends on the USER who created.
5. If a LISTING has any OFFER it generates automatically as LISTING\_ID is linked with OFFER\_ID.
6. LISTING\_STATUS\_ID tells us weather the LISTING is available or not when a USER during the time of booking.
7. Booking is made when a USER take over the LEASE.
8. Booking price depends upon the LEASE\_PERIOD.
9. USERS are of two types PREMIUM, NORMAL.
10. PREMIUM\_CUSTOMERS have advantage to view the offers before the NORMAL\_CUSTOMERS.
11. When LISTING is booked then the LISITING\_STATUS will be updated.
12. Each OFFER is linked to only that LISTING.
13. PROPERTY\_TYPE gives the information about LISTING's structure like number of beds and baths.
14. A CITY can be located in only one STATE.
15. A STATE can have more than one CITY.
16. All the LISTING bookings will be managed by MANAGEMENT.

## Security Levels at table level

Table	management_info6210	Poster_info6210	customer_info6210
<b>TAGS</b>	READ/WRITE	READ	READ
<b>SUBSCRIPTIONS</b>	READ/WRITE		READ
<b>STREET</b>	READ/WRITE	READ	READ
<b>STATE</b>	READ/WRITE	READ	READ
<b>QUERY</b>	READ/WRITE	READ/WRITE	READ/WRITE
<b>PROPERTY_TYPE</b>	READ/WRITE	READ	READ

<b>POSTER</b>	READ/WRITE	READ	
<b>OFFER</b>	READ/WRITE	READ/WRITE	READ/WRITE
<b>MANAGEMENT</b>	READ/WRITE		
<b>LISTING_TAGS</b>	READ/WRITE	READ/WRITE	READ
<b>LISTING_STATUS</b>	READ/WRITE	READ/WRITE	READ/WRITE
<b>LISTING</b>	READ/WRITE	READ/WRITE	READ
<b>LEASE_TYPE</b>	READ/WRITE	READ	READ
<b>CUSTOMER_TYPE</b>	READ/WRITE		
<b>CUSTOMER</b>	READ/WRITE		
<b>COMPLAINTS</b>	READ/WRITE		READ/WRITE
<b>CITY</b>	READ/WRITE	READ	READ
<b>AREA</b>	READ/WRITE	READ	READ
<b>ADDRESS</b>	READ/WRITE	READ	READ

## Final ER Diagram

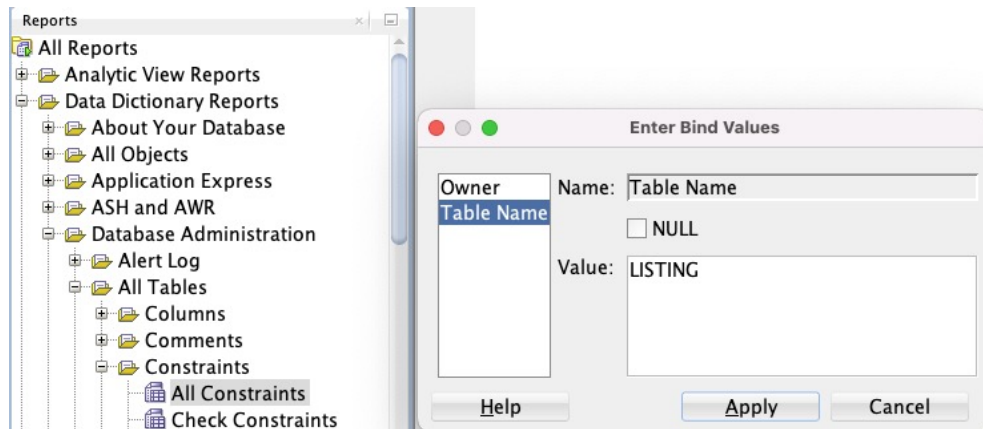


# Reports

# Reports->Data Dictionary Reports->Table->User Tables

User Tables												
Refresh: 0												
TABLE_NAME	TABLESPACE_NAME	LOGGING	NUM_ROWS	BLOCKS	EMPTY_BLOCKS	AVERAGE_ROW_LENGTH	CACHE	DATE_LAST_ANALYZED	LAST_ANALYZED	PARTITIONED	IOT_TYPE	COMPRESSION
1 ADDRESS	DATA	YES	5	5	0	11	N	24-APR-21	43.6 hours ago	NO	(null)	DISABLED
2 AREA	DATA	YES	7	5	0	10	N	24-APR-21	43.6 hours ago	NO	(null)	DISABLED
3 CITY	DATA	YES	5	5	0	17	N	24-APR-21	43.6 hours ago	NO	(null)	DISABLED
4 COMPLAINTS	DATA	YES	(null)	(null)	(null)	(null)	N	(null)	minutes ago	NO	(null)	DISABLED
5 CUSTOMER	DATA	YES	3	5	0	106	N	26-APR-21	8.1 hours ago	NO	(null)	DISABLED
6 CUSTOMER_TYPE	DATA	YES	4	5	0	13	N	24-APR-21	43.6 hours ago	NO	(null)	DISABLED
7 LEASE_TYPE	DATA	YES	5	5	0	10	N	24-APR-21	43.6 hours ago	NO	(null)	DISABLED
8 LISTING	DATA	YES	1	5	0	111	N	25-APR-21	26.7 hours ago	NO	(null)	DISABLED
9 LISTING_STATUS	DATA	YES	0	5	0	0	N	25-APR-21	26.7 hours ago	NO	(null)	DISABLED
10 LISTING_TAGS	DATA	YES	(null)	(null)	(null)	(null)	N	(null)	minutes ago	NO	(null)	DISABLED
11 MANAGEMENT	DATA	YES	1	5	0	50	N	25-APR-21	26.7 hours ago	NO	(null)	DISABLED
12 OFFER	DATA	YES	(null)	(null)	(null)	(null)	N	(null)	minutes ago	NO	(null)	DISABLED
13 POSTER	DATA	YES	1	5	0	101	N	25-APR-21	29.6 hours ago	NO	(null)	DISABLED
14 PROPERTY_TYPE	DATA	YES	5	5	0	19	N	24-APR-21	43.6 hours ago	NO	(null)	DISABLED
15 QUERY	DATA	YES	(null)	(null)	(null)	(null)	N	(null)	minutes ago	NO	(null)	DISABLED
16 STATE	DATA	YES	5	5	0	7	N	24-APR-21	43.6 hours ago	NO	(null)	DISABLED
17 STREET	DATA	YES	5	5	0	25	N	24-APR-21	43.6 hours ago	NO	(null)	DISABLED
18 SUBSCRIPTIONS	DATA	YES	2	5	0	16	N	26-APR-21	8.1 hours ago	NO	(null)	DISABLED
19 TAGS	DATA	YES	4	5	0	21	N	24-APR-21	43.6 hours ago	NO	(null)	DISABLED

# Reports->Data Dictionary Reports->Database Administration -> Constraints ->All Constraints



The screenshot shows the 'All Constraints' report for database INFO6210. The report displays 12 constraints for the LISTING table. The first constraint is a Primary Key (LISTING\_PK). The remaining 11 constraints are Foreign Keys (SYS\_C0018821 through SYS\_C0018831). All constraints are enabled.

	Owner	Table_Name	Constraint_Name	Constraint_Type	Status
1	ADMIN	LISTING	LISTING_PK	Primary_Key	Enabled
2	ADMIN	LISTING	SYS_C0018821	Check	Enabled
3	ADMIN	LISTING	SYS_C0018822	Check	Enabled
4	ADMIN	LISTING	SYS_C0018823	Check	Enabled
5	ADMIN	LISTING	SYS_C0018824	Check	Enabled
6	ADMIN	LISTING	SYS_C0018825	Check	Enabled
7	ADMIN	LISTING	SYS_C0018826	Foreign_Key	Enabled
8	ADMIN	LISTING	SYS_C0018827	Foreign_Key	Enabled
9	ADMIN	LISTING	SYS_C0018828	Foreign_Key	Enabled
10	ADMIN	LISTING	SYS_C0018829	Foreign_Key	Enabled
11	ADMIN	LISTING	SYS_C0018830	Foreign_Key	Enabled
12	ADMIN	LISTING	SYS_C0018831	Foreign_Key	Enabled

# Reports->Data Dictionary Reports->Database Administration -> Storage -> Table Count by Tablespace

Table Count by Tablespace					
INFO6210					
Refresh: 0					
	OWNER	Name	Tables	Segment Usage	Quota Usage QUOTA
1	AUDSYS	SYSAUX	33	2.0625	13.5 UNLIMITED
2	GSMADMIN_INTERNAL	SYSAUX	5	0.3125	1 UNLIMITED
3	XDB	SYSAUX	5	0.3125	0.9375 UNLIMITED
4	LBACSYS	SYSTEM	4	0.25	0.3125 UNLIMITED
5	MDSYS	SYSAUX	33	16.625	74.8125 UNLIMITED
6	C##CLOUD\$SERVICE	SYSAUX	5	0.3125	2.6875 UNLIMITED
7	ADMIN	DATA	46	2.875	6.0625 UNLIMITED
8	FLows_FILES	SYSAUX	1	0.0625	1.75 UNLIMITED
9	C##ADP\$SERVICE	SYSAUX	10	0.625	3.75 UNLIMITED
10	ORDS_METADATA	SYSAUX	9	0.5625	1.5625 UNLIMITED
11	APEX_200200	SYSAUX	285	389.6875	892.125 UNLIMITED
12	GRAPH\$METADATA	SYSAUX	1	0.0625	0.1875 UNLIMITED
13	PYQSYS	SYSTEM	70	14	14 UNLIMITED
14	SYS	SYSAUX	134	137.9375	(null) (null)
15	SSB	SAMPLESCHEMA	5	166316.9375	(null) (null)
16	SYSTEM	SYSTEM	8	0.5	(null) (null)
17	DBSNMP	SYSAUX	1	0.0625	(null) (null)
18	SYS	DBFS_DATA	2	0.125	(null) (null)
19	OML\$MODELS	SYSAUX	1	0.0625	(null) (null)
20	SH	SAMPLESCHEMA	9	5.75	(null) (null)
21	GSMADMIN_INTERNAL	SYSTEM	1	0.0625	(null) (null)
22	CTXSYS	SYSAUX	16	1.0625	(null) (null)
23	SYSTEM	SYSAUX	2	0.125	(null) (null)
24	DVSYs	SYSTEM	24	1.5	(null) (null)
25	SYS	SYSTEM	354	257.4375	(null) (null)

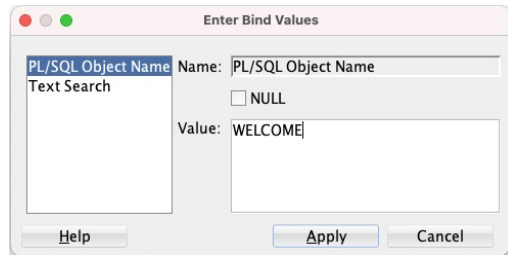


Reports->Data Dictionary Reports->Database Administration -> Sessions -> Active Session Counts

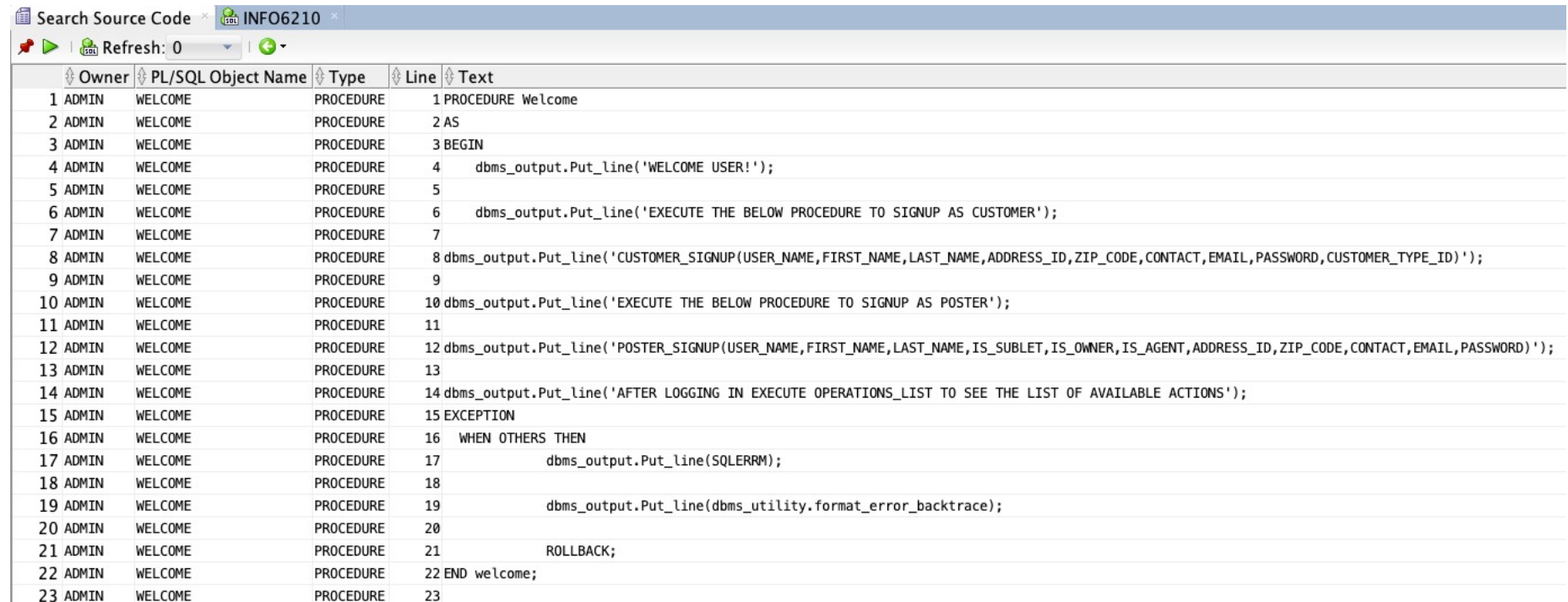


Active Session Counts		
Refresh: 0		
INST_ID	ACTIVE	TOTAL
1	1	10

# Reports->Data Dictionary Reports->Database Administration -> PLSQL -> Search Source Code



The 'Enter Bind Values' dialog box is shown. It has a title bar with standard window controls. Inside, there's a 'PL/SQL Object Name' field with the text 'PL/SQL Object Name'. Below it is a 'Text Search' section with a 'Value' field containing 'WELCOME'. There are also 'Help', 'Apply', and 'Cancel' buttons at the bottom.



The 'Search Source Code' window displays the source code for the 'WELCOME' procedure. The window title is 'Search Source Code' with a sub-tab 'INFO6210'. It includes a 'Refresh: 0' button and a search icon. The table below lists the source code lines, organized by Owner, PL/SQL Object Name, Type, Line, and Text.

Owner	PL/SQL Object Name	Type	Line	Text
ADMIN	WELCOME	PROCEDURE	1	PROCEDURE welcome
ADMIN	WELCOME	PROCEDURE	2	AS
ADMIN	WELCOME	PROCEDURE	3	BEGIN
ADMIN	WELCOME	PROCEDURE	4	dbms_output.Put_line('WELCOME USER!');
ADMIN	WELCOME	PROCEDURE	5	
ADMIN	WELCOME	PROCEDURE	6	dbms_output.Put_line('EXECUTE THE BELOW PROCEDURE TO SIGNUP AS CUSTOMER');
ADMIN	WELCOME	PROCEDURE	7	
ADMIN	WELCOME	PROCEDURE	8	dbms_output.Put_line('CUSTOMER_SIGNUP(USER_NAME,FIRST_NAME, LAST_NAME, ADDRESS_ID, ZIP_CODE, CONTACT, EMAIL, PASSWORD, CUSTOMER_TYPE_ID)');
ADMIN	WELCOME	PROCEDURE	9	
ADMIN	WELCOME	PROCEDURE	10	dbms_output.Put_line('EXECUTE THE BELOW PROCEDURE TO SIGNUP AS POSTER');
ADMIN	WELCOME	PROCEDURE	11	
ADMIN	WELCOME	PROCEDURE	12	dbms_output.Put_line('POSTER_SIGNUP(USER_NAME,FIRST_NAME, LAST_NAME, IS_SUBLET, IS_OWNER, IS_AGENT, ADDRESS_ID, ZIP_CODE, CONTACT, EMAIL, PASSWORD)');
ADMIN	WELCOME	PROCEDURE	13	
ADMIN	WELCOME	PROCEDURE	14	dbms_output.Put_line('AFTER LOGGING IN EXECUTE OPERATIONS_LIST TO SEE THE LIST OF AVAILABLE ACTIONS');
ADMIN	WELCOME	PROCEDURE	15	EXCEPTION
ADMIN	WELCOME	PROCEDURE	16	WHEN OTHERS THEN
ADMIN	WELCOME	PROCEDURE	17	dbms_output.Put_line(SQLERRM);
ADMIN	WELCOME	PROCEDURE	18	
ADMIN	WELCOME	PROCEDURE	19	dbms_output.Put_line(dbms_utility.format_error_backtrace);
ADMIN	WELCOME	PROCEDURE	20	
ADMIN	WELCOME	PROCEDURE	21	ROLLBACK;
ADMIN	WELCOME	PROCEDURE	22	END welcome;
ADMIN	WELCOME	PROCEDURE	23	

# All Reports -> User Defined Reports -> Address Views(user defined)

The screenshot shows a 'Create Report' dialog box with the following fields:

- Name:** Address Views
- Style:** Table
- Description:** All stored addresses
- SQL:**

```
SELECT "ID", "DOOR_NO", "STREET", "CITY", "STATE" FROM(
select * from address_list
)
```

Buttons at the bottom: Help, Apply, Cancel.

The screenshot shows a report window titled 'Address Views' with a tab labeled 'INFO6210'. The report displays a table with the following data:

	ID	DOOR_NO	STREET	CITY	STATE
1	1019		76 Huntington Ave	Boston	MA
2	1020		1044 Bloomfield Ave	New Jersey	NY
3	1021		12 Hollywood Boulevard	Los Angeles	CA
4	1022		98 Beresford Road	Charlotte	NC
5	1023		154 Main Street District	Dallas	TX