DENTAL HEALTH MANAGEMENT SYSTEM

GROUP 10 – EMERALD

KANCHAN CHOWDHARI

PAYAL KAUR

PURVA KHANDELWAL

SAMIKSHA MHATRE

CONTENTS

INTRODUCTION

Dental Health Management System is a web-based application which covers the essential aspects of management and operations of the dental clinic. This application will help the dentists and staff members to organize patient's dental records, schedule, view or cancel appointments, keep tabs on the treatments that have been undergone by the patients, and manage the inventory of dental supplies and equipment. This database system facilitates the retrieval of information and generates medical report for the patients and dentists to refer in the future. Additionally, it will manage the billing and printing of reports. Patients should be able to register themselves in the system by signing up using their email ID and password. Moreover, they should be able to request, view their scheduled or cancelled appointments, prescriptions, and treatment details. Dentists should be able to view and modify the patient and treatment details. The staff of the clinic should be able to organize all the patient, dentist, billing, and inventory data.

USER FUNCTIONALITY

ADMIN – They are super users, who have knowledge of the system and will be able to operate and control the entire system. They can add, delete, and modify details in the website.

DENTIST – They can view the patient details, history of medical records and appointments. They can add/ delete/modify treatment details and prescribe medicines.

ASSISTANT- They will be able to view all patient details, history of medical records and appointments.

LAB TECHNICIAN - They can view/generate the medical reports and materials needed for treatment as per dentist’s prescription.

STAFF - They can handle basic operations of the clinic and its inventory.

RECEPTIONIST - They can add/delete/modify patient details and their appointment details. They will process the billing after the treatment is done by adding various charges and collect payment from them. They can also print bills, prescriptions, and medical records.

PATIENT – They can view or modify their personal, appointment details, medical prescription and notes. They can also check available appointment dates for booking appointment and book them.

VENDORS – They can view/modify their product details.

PROJECT REQUIREMENTS

REQUIREMENTS

1. The primary focus of this project is to help the dentists and staff in managing and organizing the patient's dental records, appointment details, treatment tracking, inventories of dental supplies and equipment.
2. The database system includes tables with only essential entities: Login Details, Users, Appointments, Treatments, Billing, Orders, Products, Vendors.
3. This database should primarily support Online Transaction Processing (OLTP) Methodology.
4. To restrict user access and protect the system from unintentional or malicious destruction, security measures should be included in the system.
5. The website should facilitate the user authentication based on user role type managed in database.
6. The availability of the website is expected to be in the range of 95-98%.
7. High peaks in traffic should be tolerated by the network and server capacity.
8. In case of system failure, our system should provide standard procedures that include regularly scheduled backups and business continuity plans.
9. Appointment Reminder emails should be sent to patients two days prior to the appointment date.

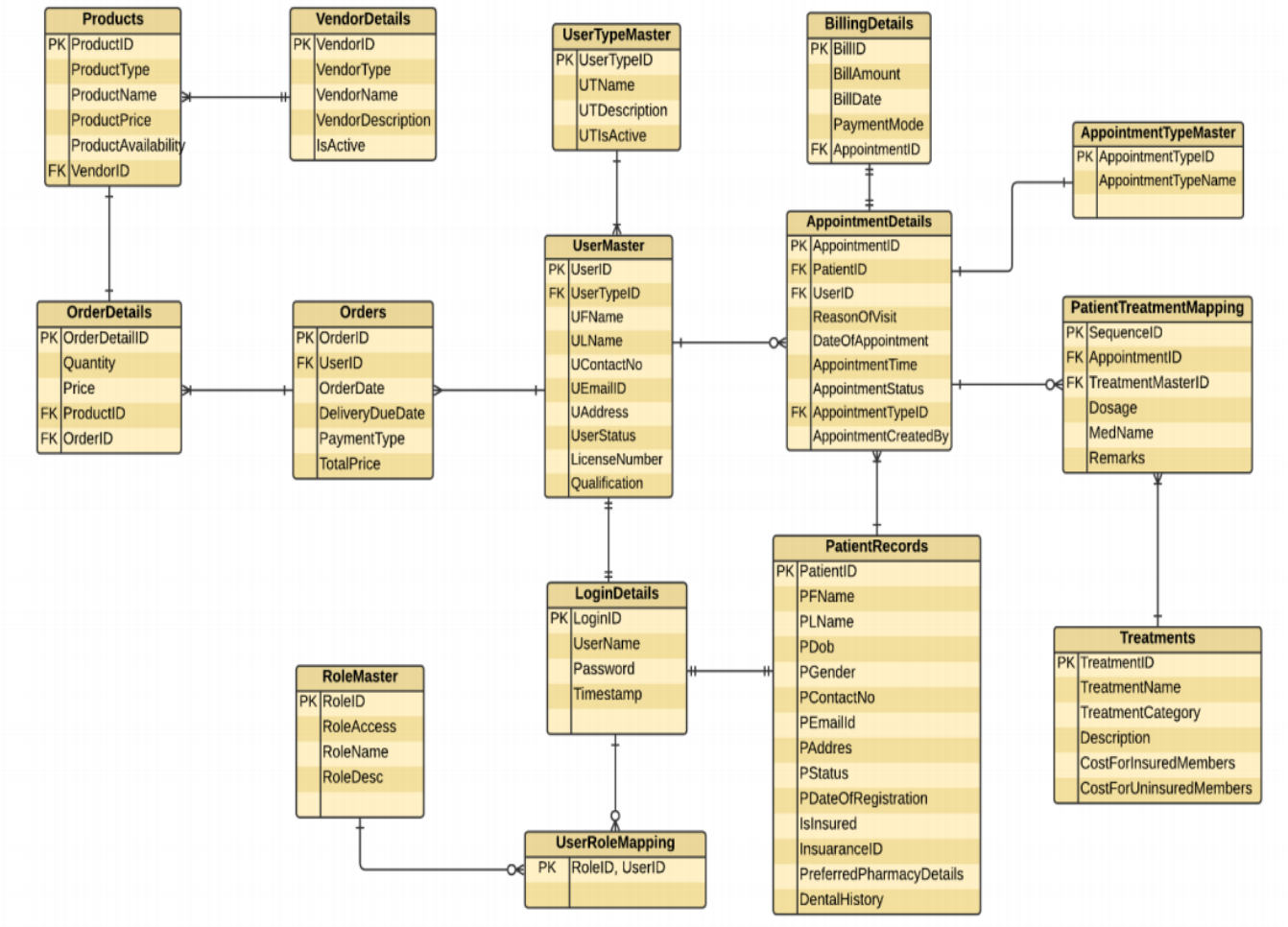
ASSUMPTIONS

1. The dental clinic database in design is for an individual clinic having single location.
2. All the users access the dental clinic system mainly via an online web interface.
3. The system can have multiple users.
4. Patients can check availability of appointments and book appointments them.
5. A patient can only book 1 appointment per day.
6. Based on demographic healthcare system, the patient is always assumed to be insured.
7. Patients are aware of the products being used in their treatments which is informed by the dentist, however they are not responsible for ordering the products needed for treatment.
8. At least one dentist is available every day.
9. A single vendor can sell multiple products.
10. One order can include multiple products.

LOGICAL DESIGN

ENTITY RELATIONSHIP DIAGRAM

ER Diagram shows relationships among various entity sets that are stored in the database.



DATA DICTIONARY

TABLES

1. **USER_TYPE_MASTER -** This table specifies the type of user working in the clinic. It can be Dentist, Staff, Admin, Assistant, Lab Technician or Receptionist.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
USER_TYPE_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
UT_NAME	VARCHAR2(100)		
IS_ACTIVE	CHAR(1)	DEFAULT 'Y'	

2. **ROLE_MASTER -** This table is used manage the access levels of different users in the system.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
ROLE_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
ROLE_ACCESS	VARCHAR2(10)		
ROLE_NAME	VARCHAR2(50)		
ROLE_DESC	VARCHAR2(100)		

3. **USER_MASTER -** This table is used to save all user details of anyone working in the clinic.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
USER_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
USER_TYPE_ID	NUMBER(10)		FOREIGN KEY
UFNAME	VARCHAR2(100)		
ULNAME	VARCHAR2(100)		
UCONTACT_NO	NUMBER(10)	NOT NULL	
UEMAIL_ID	VARCHAR2(50)		
USER_ADDRESS	VARCHAR2(500)		
USER_STATUS	VARCHAR2(10)	DEFAULT 'ACTIVE'	
LICENSE_NO			
QUALIFICATION	VARCHAR2(100)		

4. **PATIENT_RECORDS -** This table is used to save all new and existing patient details.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
PATIENT_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
PFNAME	VARCHAR2(100)		
PLNAME	VARCHAR2(100)		
P_DOB	DATE		
P_GENDER	VARCHAR2(20)		
PCONTACT_NO	NUMBER(10)		
PEMAIL_ID	VARCHAR2(50)		
P_ADDRESS	VARCHAR2(500)		
PATIENT_STATUS	VARCHAR2(10)	DEFAULT 'ACTIVE'	
DATE_OF_REGISTRATION	DATE	DEFAULT SYSDATE	
IS_INSURED	CHAR(1)	DEFAULT 'Y'	
INSUARANCE_ID	NUMBER(10)		
PREFERRED_PHARMACY	VARCHAR2(200)		
DENTAL_HISTORY	VARCHAR2(1000)		

5. **VENDOR_DETAILS -** This table stores details of all vendors supplying products and equipment.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
VENDOR_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
VENDOR_TYPE	VARCHAR2(10)		
VENDOR_NAME	VARCHAR2(100)		
VENDOR_DESC	VARCHAR2(200)		
IS_ACTIVE	CHAR(1)	DEFAULT 'Y'	

6. **USER_ROLE_MAPPING -** This table maps the user with their corresponding role.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
ROLE_ID	NUMBER(10)		
USER_ID	NUMBER(10)		

7. **LOGIN_DETAILS -** This table stores the authentication information of a user or a patient.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
LOGIN_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
USER_NAME	VARCHAR2(20)		
PASSWORD	VARCHAR2(20)		
LOGIN_DATE_TIME	TIMESTAMP		

8. **APPOINTMENT_TYPE_MASTER -** This table stores details of the appointment types i.e., New, Follow-up, Cancel.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
APPT_TYPE_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
APPT_TYPE_NAME	VARCHAR2(10)		

9. **APPOINTMENT_DTLS -** This table saves all types of appointment related details.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
APPOINTMENT_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
PATIENT_ID	NUMBER(10)		FOREIGN KEY
USER_ID	NUMBER(10)		FOREIGN KEY
REASON_OF_VISIT	VARCHAR2(100)		
APPT_DATE	DATE		
APPT_TIME	TIMESTAMP		
APPT_STATUS	VARCHAR2(10)	DEFAULT 'NEW'	
APPT_TYPE_ID	NUMBER(10)		FOREIGN KEY
APPT_CREATED_BY	NUMBER(10)	NOT NULL	

10. **TREATMENTS -** This table has the list of all available treatment and their cost of service. The treatments are categorized, for instance, a patient opting for cavity filling can choose between gold, silver and ceramic filling.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
TREATEMENT_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
TREATMENT_NAME	VARCHAR2(100)		
TREATMENT_CATEGORY	VARCHAR2(100)		
DESCRIPTION	VARCHAR2(500)		
COI_MEMBER	NUMBER (30,2)		
COI_NON_MEMBER	NUMBER (30,2)		

11. **PATIENT_TREATMENT_MAPPING -** This table stores details of treatment done on a particular patient along with its prescription and medicine details.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
PT_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
APPOINTMENT_ID	NUMBER(10)		FOREIGN KEY
TREATMENT_ID	NUMBER(10)		FOREIGN KEY
DOSAGE	VARCHAR2(100)		
MED_NAME	VARCHAR2(100)		
REMARKS	VARCHAR2(500)		

12. **BILLING_DETAILS -** This table stores the payment details for the patient.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
BIIL_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
BILL_AMOUNT	FLOAT(10,2)	DEFAULT 0	
BILL_DATE	DATE	DEFAULT SYSDATE	
PAYMENT_MODE	VARCHAR2(10)		
APPOINTMENT_ID	NUMBER(10)		

13. **PRODUCTS -** This table stores the product & equipment details required for dental operations.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
PRODUCT_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
PRODUCT_TYPE	VARCHAR2(30)		
PRODUCT_NAME	VARCHAR2(100)		
PRODUCT_PRICE	NUMBER (8,4)		
PROD_AVAILABILITY	VARCHAR2(10)		
VENDOR_ID	NUMBER(10)		FOREIGN KEY

14. **ORDER_DETAILS -** This table stores order details of various products and equipment required for dental operations.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
ORDER_DTL_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
QUANTITY	NUMBER(10)		
PRICE	FLOAT(10,2)	DEFAULT 0	
ORDER_ID	NUMBER(10)		FOREIGN KEY
PRODUCT_ID	NUMBER(10)		FOREIGN KEY

15. **ORDERS -** This table stores the header details of the orders generated.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
ORDER_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
ORDER_DATE	DATE		
DELIVERY_DUE_DATE	DATE		
PAYMENT_MODE	VARCHAR2(10)		
TOTAL_PRICE	FLOAT(10,2)	DEFAULT 0	
USER_ID	NUMBER(10)		FOREIGN KEY

SEQUENCES

Syntax for creating Sequence:

```
CREATE SEQUENCE APPT_ID_SEQ
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE
```

The different sequences used in the project are as listed below:

1. APPT_ID_SEQ
2. APPT_TYPE_SEQ
3. BILL_SEQ
4. ORDER_DTL_SEQ
5. ORDER_SEQ
6. PAT_ID_SEQ
7. PRODUCT_SEQ
8. ROLE_ID_SEQ
9. TREATMENT_SEQ
10. USER_ID_SEQ
11. USER_TYPE_SEQ
12. VENDOR_SEQ

CONSTRAINTS

Syntax for creating Primary key and Foreign key:

1. ALTER TABLE user_type_master
   ADD CONSTRAINT Utm_pk PRIMARY KEY(user_type_id);
2. ALTER TABLE user_master
   ADD CONSTRAINT User_master_pk PRIMARY KEY(user_id);
3. ALTER TABLE user_master
   ADD CONSTRAINT User_Master_FK FOREIGN KEY(user_type_id)
   REFERENCES User_type_master (user_type_id);

Below table shows all the constraints created in the system.

	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	R_CONSTRAINT_NAME
1	APPOINTMENT_DTLS_FK1	R	APPOINTMENT_DTLS	PATIENT_RECORDS_PK
2	APPOINTMENT_DTLS_FK2	R	APPOINTMENT_DTLS	USER_MASTER_PK
3	APPOINTMENT_DTLS_FK3	R	APPOINTMENT_DTLS	APPOINTMENT_TYPE_MASTER_PK
4	APPOINTMENT_DTLS_PK	P	APPOINTMENT_DTLS	(null)
5	APPOINTMENT_TYPE_MASTER_PK	P	APPOINTMENT_TYPE_MASTER	(null)
6	BILLING_DTLS_PK	P	BILLING_DETAILS	(null)
7	TABLE1_PK	P	LOGIN_DETAILS	(null)
8	ORD_FK	R	ORDERS	USER_MASTER_PK
9	ORD_PK	P	ORDERS	(null)
10	OD_FK1	R	ORDER_DETAILS	PRODUCT_ID_PK
11	OD_FK2	R	ORDER_DETAILS	ORD_PK
12	OD_PK	P	ORDER_DETAILS	(null)
13	PATIENT_RECORDS_PK	P	PATIENT_RECORDS	(null)
14	PATIENT_TREATMENT_MAPPING_FK2	R	PATIENT_TREATMENT_MAPPING	TREATMENTS_PK
15	PATIENT_TREATMENT_MAPPING_PK	P	PATIENT_TREATMENT_MAPPING	(null)
16	PATIENT_TREATMENT_MAP_FK1	R	PATIENT_TREATMENT_MAPPING	APPOINTMENT_DTLS_PK
17	PRODUCTS_FK	R	PRODUCTS	VENDOR_DETAILS_PK
18	PRODUCT_ID_PK	P	PRODUCTS	(null)
19	RM_PK	P	ROLE_MASTER	(null)
20	TREATMENTS_PK	P	TREATMENTS	(null)
21	USER_MASTER_FK	R	USER_MASTER	UTM_PK
22	USER_MASTER_PK	P	USER_MASTER	(null)
23	URM_PK	P	USER_ROLE_MAPPING	(null)
24	UTM_PK	P	USER_TYPE_MASTER	(null)
25	VENDOR_DETAILS_PK	P	VENDOR_DETAILS	(null)

PHYSICAL DESIGN

The Dental Clinic Database System is a relational database consisting of Online Transaction Processing (OLTP) for dental operations and services. This database system is confined to a small-scale environment which supports the features of high concurrency and throughput. The dental clinic database will store historic and current data to facilitate ad-hoc querying, preparing significant reports for administrative purposes, and allow data mining related activities for future analysis and requirements.

This system will include basic database objects like tables, indices, integrity constraints and subroutines such as stored procedures. It includes 15 tables, few of those are- appointment_details, patient_records, treatments, products, vendors, etc. We have designed the database by normalizing the table values up to 3NF. In order to provide smaller response time and outputs, we have proposed the functionality of archiving the data once in every 3 months. On account of the inclusion of indexes, we would expedite the query transaction processing.

Database administrative activities would be monitored regularly by running DBA scripts for determining current, active and inactive database sessions, storage statistics, queries that consume the most resources and sending notification reminders via email using automated procedures.

DATA GENERATION & LOADING

After creation of data structures insert scripts, procedures to generate data and import function through excel available in Oracle SQL Developer was used to generate relevant data. This method for generating data was fast and accurate with some time spent on gathering the relevant information from the internet. We generated limited records for development purposes, count of each table are as below:

TABLE	COUNT
APPOINTMENT_TYPE_MASTER	3
APPOINTMENT_DTLS	10000
BILLING_DETAILS	200
LOGIN_DETAILS	4163
ORDERS	200
ORDER_DETAILS	200
PATIENT_RECORDS	4135
PATIENT_TREATMENT_MAPPING	3447
PRODUCTS	502
ROLE_MASTER	4
TREATMENTS	35
USER_TYPE_MASTER	6
USER_MASTER	22
USER_ROLE_MAPPING	4287
VENDOR_DETAILS	136

USER_TYPE_MASTER

```
insert into USER_TYPE_MASTER values (1,'Admin','Y);
insert into USER_TYPE_MASTER values (user_type_seq.nextval,'Dentist','Y);
insert into USER_TYPE_MASTER values (user_type_seq.nextval,'Assistant','Y);
insert into USER_TYPE_MASTER values (user_type_seq.nextval,'Lab Technician','Y);
insert into USER_TYPE_MASTER values (user_type_seq.nextval,'Staff','Y);
insert into USER_TYPE_MASTER values (user_type_seq.nextval,'Receptionist','Y);
```

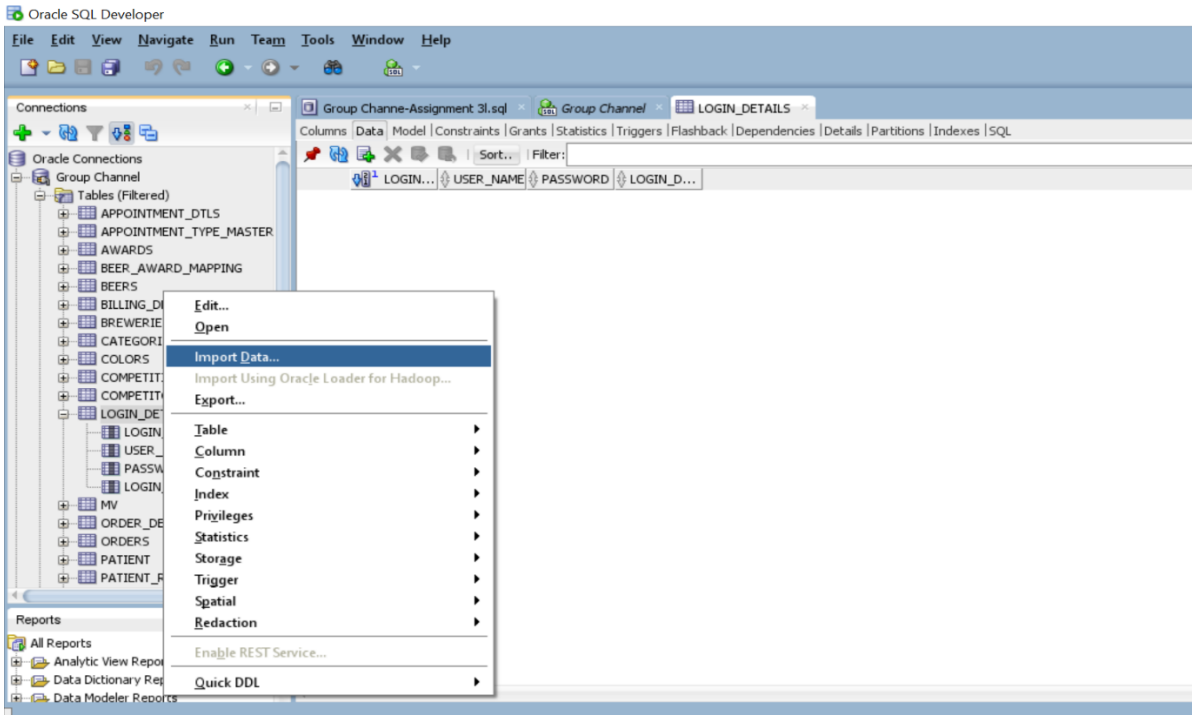
ROLE_MASTER

```
insert into role_master values (1,'ADMIN','ADMIN ROLE','This role will have all the permissions in the system ');
insert into role_master values (2,'USER','USER ROLE','This role will be assigned to anyone working in the clinic
');
insert into role_master values (3,'PATIENT','PATIENT ROLE ','This will be assigned to all the patient registered
in the system');
insert into role_master values (4,'VENDOR','VENDOR ROLE','This will be assigned to all the vendors registered
in the system');
```

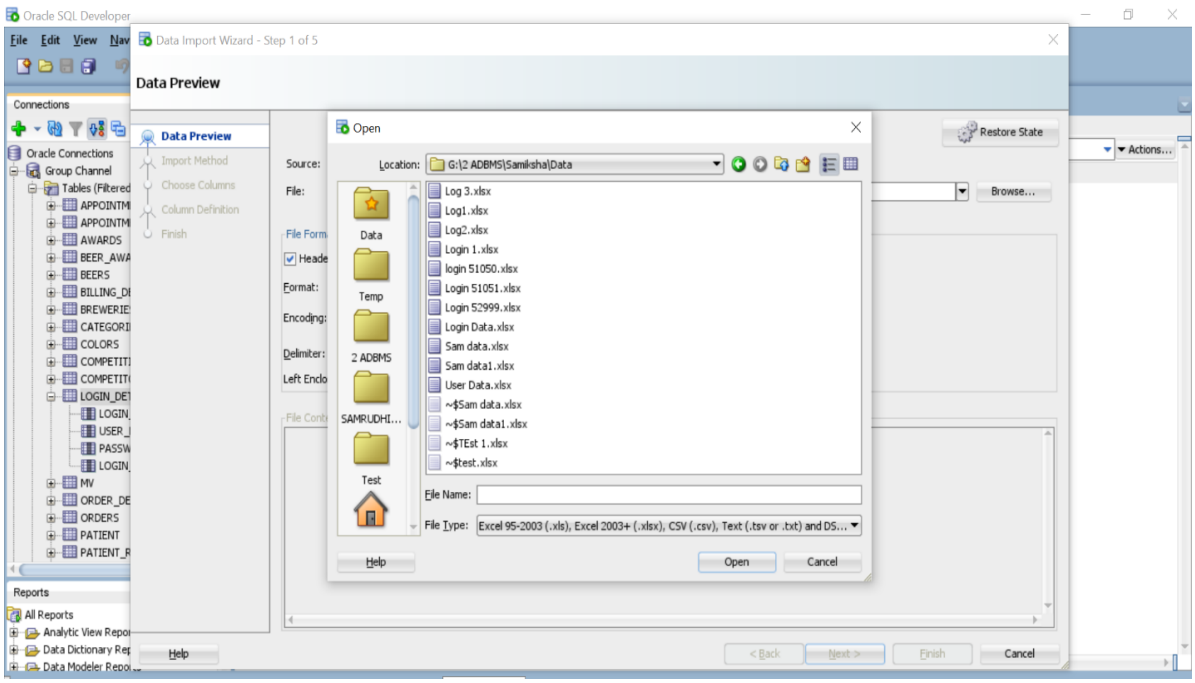
LOGIN_DETAILS

Following steps are performed to load the data.

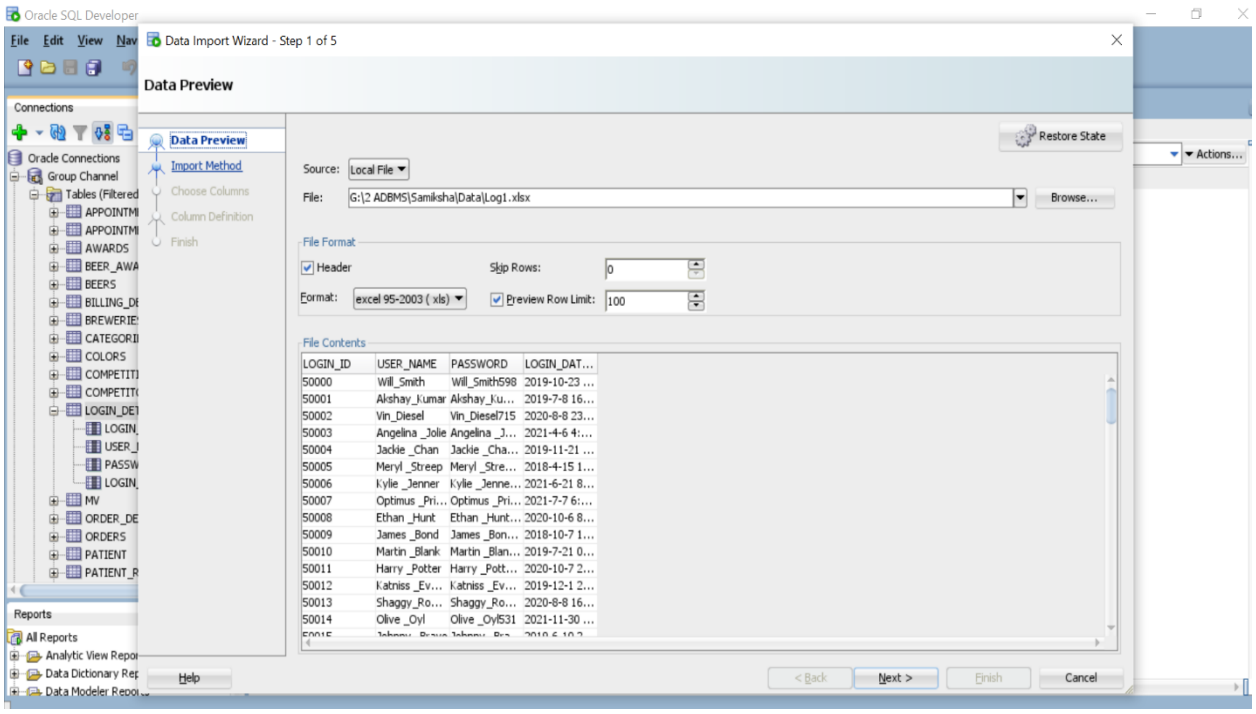
i. Right click on the table name and click on Import Data option



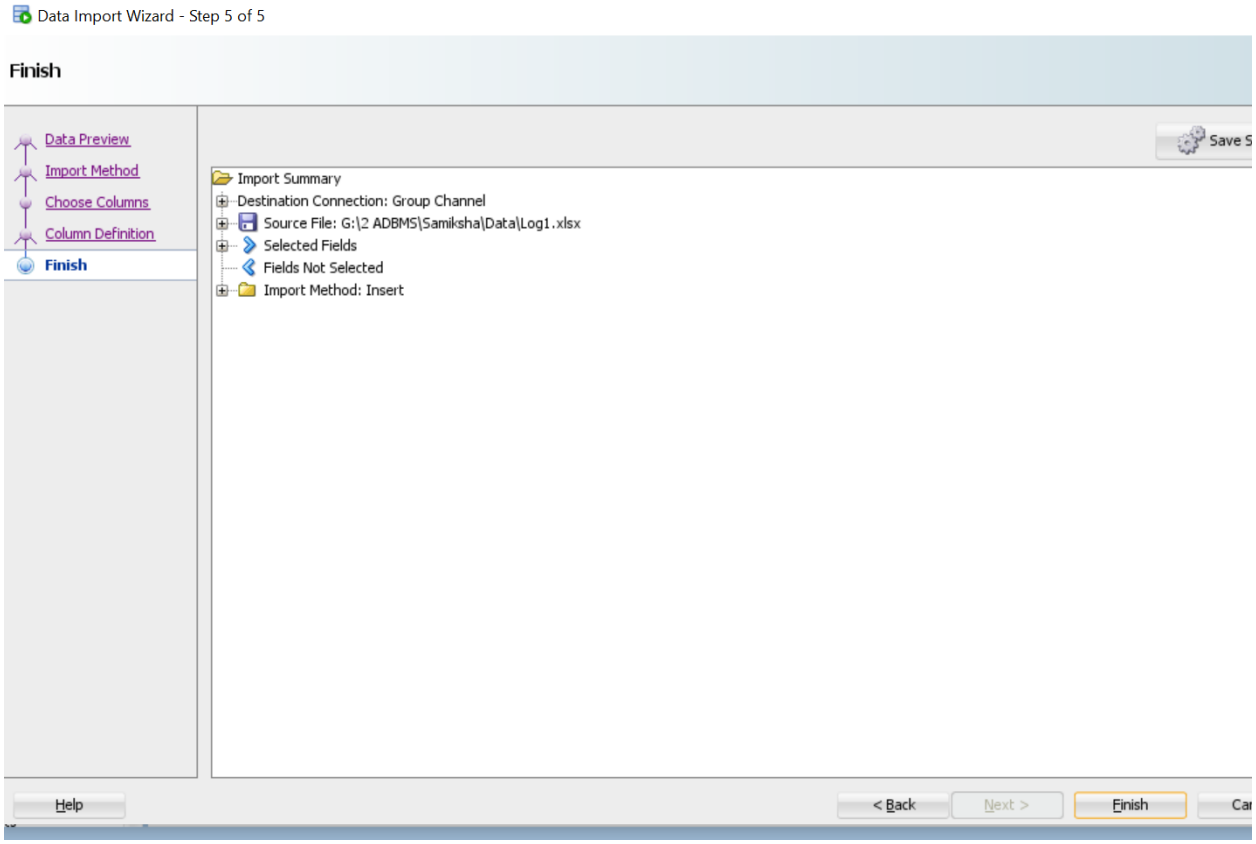
ii. Browse for the excel data sheet to be uploaded

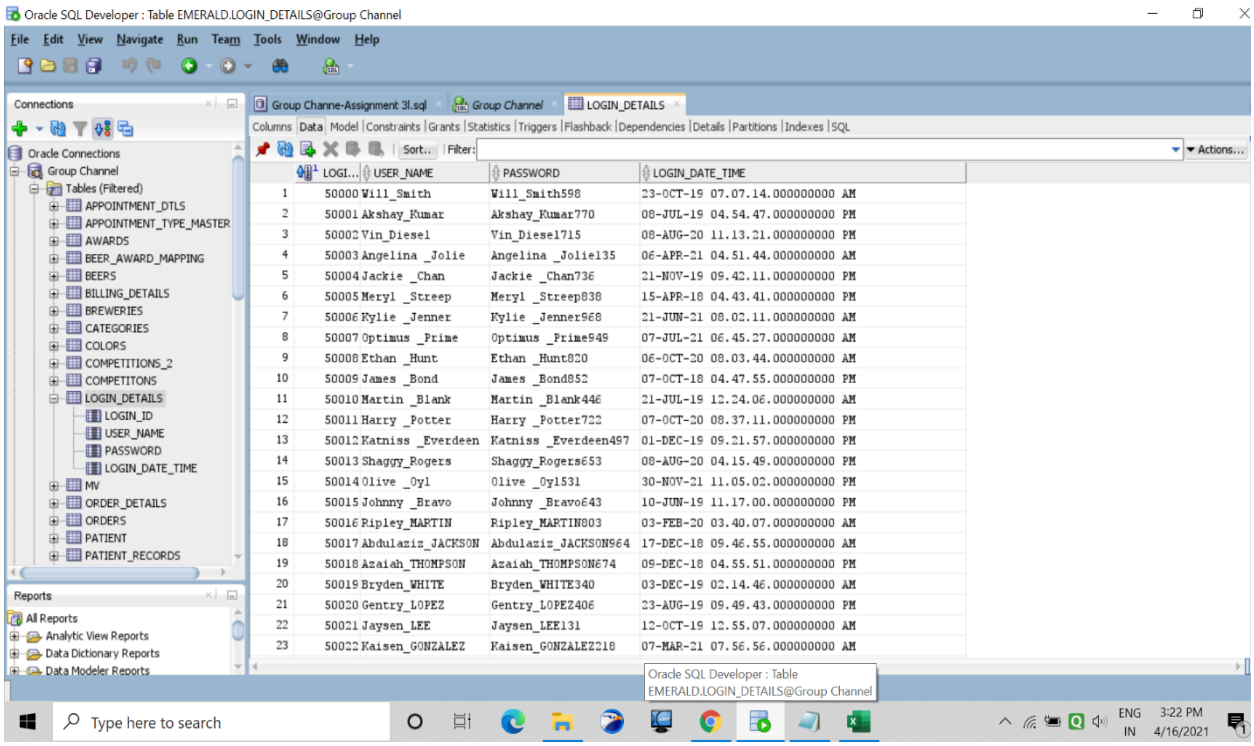


iii. Preview of the Data Sheet



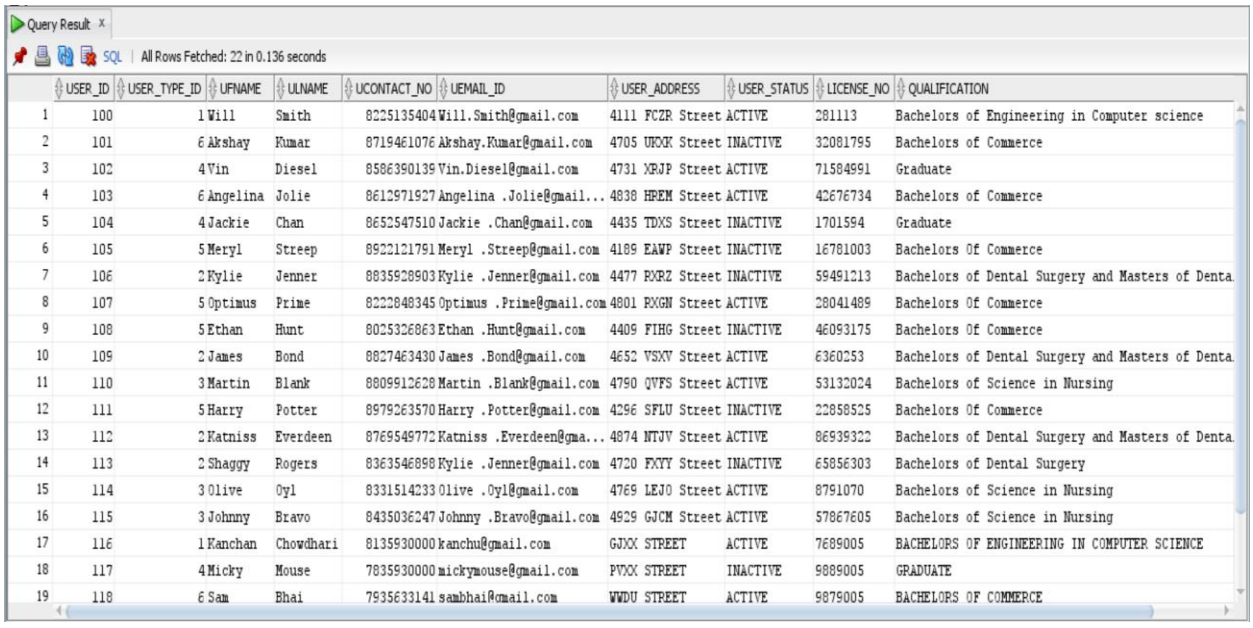
iv. After clicking on Next and Finish the data will be loaded in the table





USER_MASTER

The User_Master table data has been inserted in the same manner as the Login_Details data i.e. by importing an excel sheet. Below is the screenshot of the inserted data.



Query Result

All Rows Fetched: 22 in 0.136 seconds

	USER_ID	USER_TYPE_ID	UFNAME	ULNAME	UCONTACT_NO	UEMAIL_ID	USER_ADDRESS	USER_STATUS	LICENSE_NO	QUALIFICATION
1	100	1	Will	Smith	8225135404	Will.Smith@gmail.com	4111 FCZR Street	ACTIVE	281113	Bachelors of Engineering in Computer science
2	101	6	Akshay	Kumar	8719461076	Akshay.Kumar@gmail.com	4705 UKXK Street	INACTIVE	32081795	Bachelors of Commerce
3	102	4	Vin	Diesel	8586390139	Vin.Diesel@gmail.com	4731 XRJP Street	ACTIVE	71584991	Graduate
4	103	6	Angelina	Jolie	8612971927	Angelina .Jolie@gmail...	4838 HREM Street	ACTIVE	42676734	Bachelors of Commerce
5	104	4	Jackie	Chan	8652547510	Jackie .Chan@gmail.com	4435 TDXS Street	INACTIVE	1701594	Graduate
6	105	5	Meryl	Streep	8922121791	Meryl .Streep@gmail.com	4189 EAWP Street	INACTIVE	16781003	Bachelors Of Commerce
7	106	2	Kylie	Jenner	8835928903	Kylie .Jenner@gmail.com	4477 PXRZ Street	INACTIVE	59491213	Bachelors of Dental Surgery and Masters of Denta
8	107	5	Optimus	Prime	8222848345	Optimus .Prime@gmail.com	4801 PXGN Street	ACTIVE	28041489	Bachelors Of Commerce
9	108	5	Ethan	Hunt	8025326863	Ethan .Hunt@gmail.com	4409 FIHG Street	INACTIVE	46093175	Bachelors Of Commerce
10	109	2	James	Bond	8827463430	James .Bond@gmail.com	4652 VSXV Street	ACTIVE	6360253	Bachelors of Dental Surgery and Masters of Denta
11	110	3	Martin	Blank	8809912628	Martin .Blank@gmail.com	4790 QVFS Street	ACTIVE	53132024	Bachelors of Science in Nursing
12	111	5	Harry	Potter	8979263570	Harry .Potter@gmail.com	4296 SFLU Street	INACTIVE	22858525	Bachelors Of Commerce
13	112	2	Katniss	Everdeen	8769549772	Katniss .Everdeen@gma...	4874 NTJV Street	ACTIVE	86939322	Bachelors of Dental Surgery and Masters of Denta
14	113	2	Shaggy	Rogers	8363546898	Kylie .Jenner@gmail.com	4720 FXYY Street	INACTIVE	65856303	Bachelors of Dental Surgery
15	114	3	Olive	Oyl	8331514233	Olive .Oyl@gmail.com	4769 LEJO Street	ACTIVE	8791070	Bachelors of Science in Nursing
16	115	3	Johnny	Bravo	8435036247	Johnny .Bravo@gmail.com	4929 GJCM Street	ACTIVE	57867605	Bachelors of Science in Nursing
17	116	1	Kanchan	Chowdhari	8135930000	kanchu@gmail.com	GJXX STREET	ACTIVE	7689005	BACHELORS OF ENGINEERING IN COMPUTER SCIENCE
18	117	4	Micky	Mouse	7835930000	mickymouse@gmail.com	PVXX STREET	INACTIVE	9889005	GRADUATE
19	118	6	Sam	Bhai	7935633141	sambhai@gmail.com	WWDU STREET	ACTIVE	9879005	BACHELORS OF COMMERCE

PATIENT_RECORDS

The Patient_Records table data has been inserted in the same manner as the Login_Details data i.e. by importing an excel sheet. Below is the screenshot of the inserted data.

	PATIENT_ID	PFNAME	PLNAME	P_DOB	P_GENDER	PCONTACT_NO	PEMAIL_ID	P_ADDRESS	PATIENT_STATU
1	1085	Hannah	Myers	13-APR-16	F	2837525266	Hannah.MYERS@gmail.com	CURRIE & GRIBBEN SURGERY	INACTIVE
2	1086	Aria	Long	01-JAN-97	F	2866322983	Aria.LONG@gmail.com	18 DARLING STREET	INACTIVE
3	1087	Layla	Foster	04-OCT-96	F	2844612231	Layla.FOSTER@gmail.com	35 ST PATRICKS AVENUE	ACTIVE
4	1088	Colton	Sanders	01-FEB-97	M	2828260797	Colton.SANDERS@gmail.com	11 VICTORIA ROAD	INACTIVE
5	1089	Brooklyn	Ross	01-JAN-97	F	2886762206	Brooklyn.ROSS@gmail.com	12 LOY STREET	ACTIVE
6	1090	Jordan	Morales	05-OCT-96	M	2825652044	Jordan.MORALES@gmail.com	PENTAGON HOUSE	INACTIVE
7	1091	Brayden	Powell	01-JAN-97	M	2890457561	Brayden.POWELL@gmail.com	266 WOODSTOCK ROAD	ACTIVE
8	1092	Nicholas	Sullivan	05-OCT-96	M	2890860565	Nicholas.SULLIVAN@gmail.com	50 STATION ROAD	INACTIVE
9	1093	Robert	Russell	04-FEB-97	M	2891270041	Robert.RUSSELL@gmail.com	32 ABBEY STREET	INACTIVE
10	1094	Angel	Ortiz	07-JAN-97	M	2828260170	Angel.ORTIZ@gmail.com	14 VICTORIA ROAD	ACTIVE
11	1095	Alexa	Jenkins	06-OCT-96	F	2892663222	Alexa.JENKINS@gmail.com	29 WALLACE AVENUE	ACTIVE
12	1096	Hudson	Gutierrez	04-FEB-97	M	2882249555	Hudson.GUTIERREZ@gmail.com	3 JAMES STREET	ACTIVE
13	1097	Zoe	Perry	07-JAN-97	F	2837524958	Zoe.PERRY@gmail.com	12/14 RUSSELL STREET	ACTIVE
14	1098	Lincoln	Butler	07-OCT-96	M	2887784977	Lincoln.BUTLER@gmail.com	28 DUNGANNON STREET	ACTIVE
15	1099	Penelope	Barnes	04-FEB-97	F	2891270041	Penelope.BARNES@gmail.com	32 ABBEY STREET	INACTIVE
16	1100	Evan	Fisher	07-JAN-97	M	2890483240	Evan.FISHER@gmail.com	1003 UPPER NEWTOWNARDS RD	ACTIVE
17	1101	Dominic	Henderson	08-OCT-96	M	2825639333	Dominic.HENDERSON@gmail.com	13-15 DUKE STREET	ACTIVE

APPOINTMENT_TYPE_MASTER

insert into APPOINTMENT_TYPE_MASTER values (appt_type_seq.nextval,'NEW);
insert into APPOINTMENT_TYPE_MASTER values (appt_type_seq.nextval,'FOLLOWUP);
insert into APPOINTMENT_TYPE_MASTER values (appt_type_seq.nextval,'CANCEL);

APPOINTMENT_DTLS

The Appointment_Dtls table data has been inserted in the same manner as the Login_Details data i.e. by importing an excel sheet. Below is the screenshot of the inserted data.

	APPOINTMENT_ID	PATIENT_ID	USER_ID	REASON_OF_VISIT	APPT_DATE	APPT_TIME	APPT...	...	APPT_CREATED_BY
1	85	1221	106	039 - EXTRACRANIAL PROCEDURES W/O...	26-SEP-16	6:00:00 PM	NEW	1	103
2	86	4852	106	039 - EXTRACRANIAL PROCEDURES W/O...	11-OCT-16	7:00:00 PM	NEW	1	103
3	87	3467	106	039 - EXTRACRANIAL PROCEDURES W/O...	10-NOV-16	8:00:00 PM	NEW	1	103
4	88	4494	106	039 - EXTRACRANIAL PROCEDURES W/O...	26-SEP-16	9:00:00 PM	NEW	1	103
5	89	3940	106	039 - EXTRACRANIAL PROCEDURES W/O...	11-OCT-16	9:00:00 AM	NEW	1	103
6	90	4445	106	039 - EXTRACRANIAL PROCEDURES W/O...	11-OCT-16	10:00:00 AM	NEW	1	103
7	91	4135	106	039 - EXTRACRANIAL PROCEDURES W/O...	26-SEP-16	11:00:00 AM	NEW	1	103
8	92	2979	106	039 - EXTRACRANIAL PROCEDURES W/O...	26-SEP-16	12:00:00 PM	NEW	1	103
9	93	3637	106	039 - EXTRACRANIAL PROCEDURES W/O...	26-SEP-16	1:00:00 PM	NEW	1	103
10	94	1128	106	039 - EXTRACRANIAL PROCEDURES W/O...	26-SEP-16	2:00:00 PM	NEW	1	103
11	95	2337	106	039 - EXTRACRANIAL PROCEDURES W/O...	26-OCT-16	3:00:00 PM	NEW	1	103
12	96	2227	106	039 - EXTRACRANIAL PROCEDURES W/O...	25-SEP-16	4:00:00 PM	NEW	1	103
13	97	3418	106	039 - EXTRACRANIAL PROCEDURES W/O...	11-OCT-16	5:00:00 PM	NEW	1	103
14	98	2369	106	039 - EXTRACRANIAL PROCEDURES W/O...	11-OCT-16	6:00:00 PM	NEW	1	103
15	99	2478	106	039 - EXTRACRANIAL PROCEDURES W/O...	26-SEP-16	7:00:00 PM	NEW	1	103
16	100	2491	106	039 - EXTRACRANIAL PROCEDURES W/O...	11-OCT-16	8:00:00 PM	NEW	1	103
17	101	1093	106	039 - EXTRACRANIAL PROCEDURES W/O...	11-OCT-16	9:00:00 PM	NEW	1	103
18	102	1221	106	039 - EXTRACRANIAL PROCEDURES W/O...	26-SEP-16	9:00:00 AM	NEW	1	103

TREATMENTS

The Treatments table data has been inserted in the same manner as the Login_Details data i.e. by importing an excel sheet. Below is the screenshot of the inserted data.

	TREATEMENT_ID	TREATMENT_NAME	TREATMENT_CATEGORY	DESCRIPTION	COI_MEMBER	COI_NON_MEMBER
1	1	Craniofacial defects	Low	Craniofacia...	3764	5777
2	2	Craniofacial defects	Average	Craniofacia...	4864	5877
3	3	Craniofacial defects	High	Craniofacia...	5964	6977
4	4	Dental caries (tooth decay)	Carries Removal	Dental cari...	4977	5788
5	5	Dental caries (tooth decay)	Tooth Extraction	Dental cari...	2477	3788
6	6	Dental caries (tooth decay)	Dental Filling	Dental cari...	6977	7788
7	7	Dental sealants	(null)	Dental seal...	4454	5435
8	8	facial pain	Low Pain	The most co...	8129	9418
9	9	facial pain	High Pain	The most co...	7129	10418
10	10	facial pain	Extremely High Pain	The most co...	8129	9418
11	11	Oral Cancer	1st Stage	Approximate...	9951	10658
12	12	Oral Cancer	2nd Stage	Approximate...	8851	10658
13	13	Oral Cancer	3rd Stage	Approximate...	9051	13658
14	14	Periodontal disease	Moderate	Overall, th...	8374	9654
15	15	Periodontal disease	Severe	Overall, th...	10374	11654
16	16	Tooth loss	(null)	Tooth loss ...	4761	5835
17	17	National Health and Nutrition Examination	(null)	There are s...	5859	8031
18	18	Water Fluoridation	(null)	Since commu...	5228	6113
19	19	Dental caries (tooth decay) 2-11 years	(null)	Caries has ...	4387	5541

PATIENT_TREATMENT_MAPPING

The Patient_Treatment_mapping table data has been inserted in the same manner as the Login_Details data i.e. by importing an excel sheet. Below is the screenshot of the inserted data.

	PT_ID	APPOINTMENT_ID	TREATMENT_ID	DOSAGE	MED_NAME
1	1245	246	19	Once a day only morning before eating	Antoxines multivit 100mg capsule
2	1246	247	7	Once a day only morning before eating	Klynx Gel 10gm
3	1247	248	2	Once a day only morning before eating	oraflora 50mg tablets
4	1248	249	12	Once a day only morning before eating	Lacne Foaming Face Wash Lotion 60ml
5	1249	250	10	Once a day only morning before eating	LEE Neem Face Wash 70gm
6	1250	251	19	Once a day only morning before eating	Antoxines multivit 100mg capsule
7	1251	252	4	Once a day only morning before eating	Luminosa M Cream 20gm
8	1252	253	5	Once a day only morning before eating	Medapine AC Gel 15gmMedapine 1percent Gel
9	1253	254	15	Once a day only morning before eating	Medapine AC Gel 15gmMedapine 1percent Gel 15gm
10	1254	255	19	Once a day only morning before eating	Antoxines multivit 100mg capsule
11	1255	256	13	Once a day only morning before eating	Medsop Soap 100gmMedsop Soap 75gmMedsop Acne Soap 75gm
12	1256	257	10	Once a day only morning before eating	Melacare HC Cream 15gm
13	1257	258	13	Once a day only morning before eating	Melagard Tube 1'S
14	1258	259	17	Once a day only morning before eating	Melalite Plus Cream 15gm
15	1259	260	18	Once a day only morning before eating	Melalong AD Cream 15gm
16	1260	261	4	Once a day only morning before eating	MELANORM MS Cream 15gm
17	1261	262	11	Once a day only morning before eating	Melasol Cream 15gm

BILLING_DETAILS

```
create or replace PROCEDURE PROC_DATA_GEN_BILL_DTLS
AS
BEGIN
    FOR i IN 1..200
       LOOP
            INSERT INTO BILLING_DETAILS (BIIL_ID
                                ,BILL_AMOUNT
                                ,BILL_DATE
                                ,PAYMENT_MODE
                                ,APPOINTMENT_ID)
                        VALUES (BILL_SEQ.NEXTVAL
                                ,round(dbms_random.value(50,1000))
                                ,SYSDATE
                                ,'ONLINE'
                                ,round(dbms_random.value(1,1901)));
            COMMIT;
```

UPDATE billing_details SET bill_date = bill_date - dbms_random.value(1,1901) WHERE biil_id BETWEEN 1 AND 10;
UPDATE billing_details SET payment_mode ='CASH' WHERE bill_amount LIKE '%1%';

	BIIL_ID	BILL_AMOUNT	BILL_DATE	PAYMENT_MODE	APPOINTMENT_ID
1	1	15840	27-SEP-16	CASH	1059
2	2	51710	26-SEP-16	ONLINE	35
3	3	27850	11-OCT-16	CASH	79
4	4	25990	26-SEP-16	CASH	891
5	5	30430	11-SEP-16	CASH	362
6	6	20660	27-SEP-16	CASH	1793
7	7	70850	11-SEP-16	ONLINE	1476
8	8	44360	27-DEC-16	ONLINE	448
9	9	57290	11-SEP-16	ONLINE	1437
10	10	92380	09-SEP-16	ONLINE	1443
11	11	74240	10-SEP-16	ONLINE	1349
12	12	63990	10-SEP-16	ONLINE	1362
13	13	22630	04-AUG-17	ONLINE	668
14	14	52700	08-DEC-16	ONLINE	429
15	15	84650	10-JUN-17	CASH	613
16	16	20000	09-NOV-16	ONLINE	1870
17	17	37030	16-JUN-17	ONLINE	619
18	18	83670	26-SEP-16	ONLINE	1305

VENDOR_DETAILS

Insert into VENDOR_DETAILS (VENDOR_ID, VENDOR_TYPE, VENDOR_NAME, VENDOR_DESC, IS_ACTIVE) values (1001,'Surgery','Mydent International','t produces over half of the consumables used in dental surgery and related work','N');
Insert into VENDOR_DETAILS (VENDOR_ID, VENDOR_TYPE, VENDOR_NAME, VENDOR_DESC, IS_ACTIVE) values (1002,'dental equipment','Metrex','Metrex specializes in moderately priced, high-quality dental equipment','N');
Insert into VENDOR_DETAILS (VENDOR_ID, VENDOR_TYPE, VENDOR_NAME, VENDOR_DESC, IS_ACTIVE) values (1003,'dental equipment','Carestream','Carestream tops the list of dental equipment suppliers focused on dental technology','Y');
Insert into VENDOR_DETAILS (VENDOR_ID, VENDOR_TYPE, VENDOR_NAME, VENDOR_DESC, IS_ACTIVE) values (1004,'restoratives','Kerr','Known for superb customer service, the company offers resources related to dental diagnosis, restoratives','Y');
Insert into VENDOR_DETAILS (VENDOR_ID, VENDOR_TYPE, VENDOR_NAME, VENDOR_DESC, IS_ACTIVE) values (1005,'Preventives','DMG America','DMG is famous for its emphasis on efficiency in dental operation','Y');

Inserted records for another 129 records with the similar insert script.

	VENDOR_ID	VENDOR_TYPE	VENDOR_NAME	VENDOR_DESC	IS_ACTIVE
1	1001	Surgery	Mydent International	t produces over half of the c...	N
2	1002	Dental Equipments	Metrex	Metrex specializes in moderat...	N
3	1003	Dental Equipments	Carestream	Carestream tops the list of d...	Y
4	1004	Restoratives	Kerr	Known for superb customer ser...	Y
5	1005	Preventives	DMG America	DMG is famous for its emphasi...	Y
6	1006	Dental Equipments	3M Oral Care	It holds a secure place among...	Y
7	1007	Restoratives	Sultan Healthcare	Sultan specializes in top-qua...	Y
8	1008	Restoratives	Septodont	It is a world-wide distribut...	Y
9	1009	Dental Equipments	Patterson Dental Supplies	Patterson is known for its br...	Y
10	1010	Digital Technology	Midwest Dental Supply	Midwest specializes in X-ray ...	Y
11	1011	Endodontists	Pearson Dental Supplies	Pearson caters primarily to s...	N
12	1012	Dental Equipments	Darby Dental Supplies	Darby is famous for its massi...	N
13	1013	Digital Technology	Henry Schein	Schein boasts more than 1 mil...	Y
14	1014	Dental Equipments	Benco Dental Supplies	Benco has been a key player i...	Y
15	1015	Dental Equipments	ABrite Dental	ABrite Dental Inc is a dedica...	Y
16	1016	Dental Equipments	Ace Dental Supply	specialize in offering quick...	Y
17	1017	Dental Equipments	AdDent	AdDent, Inc. is a Dental Manu...	Y
18	1018	Dental Equipments	ADS, Inc.	ADS, Inc. is a family owned e...	Y

PRODUCTS

Insert into PRODUCTS (PRODUCT_ID, PRODUCT_TYPE, PRODUCT_NAME, PRODUCT_PRICE, PROD_AVAILABILITY,VENDOR_ID) values (214,'Endodontics','#6 hand files',129.195,'NO',1105);
Insert into PRODUCTS (PRODUCT_ID, PRODUCT_TYPE, PRODUCT_NAME, PRODUCT_PRICE, PROD_AVAILABILITY,VENDOR_ID) values (215,'Endodontics','#8 hand files',118.5079,'NO',1105);
Insert into PRODUCTS (PRODUCT_ID, PRODUCT_TYPE, PRODUCT_NAME, PRODUCT_PRICE, PROD_AVAILABILITY,VENDOR_ID) values (216,'Endodontics','#10 hand files',119.8855,'NO',1129);
Insert into PRODUCTS (PRODUCT_ID, PRODUCT_TYPE, PRODUCT_NAME, PRODUCT_PRICE, PROD_AVAILABILITY,VENDOR_ID) values (217,'Endodontics','#15 hand files',115.1252,'NO',1085);
Insert into PRODUCTS (PRODUCT_ID, PRODUCT_TYPE, PRODUCT_NAME, PRODUCT_PRICE, PROD_AVAILABILITY,VENDOR_ID) values (218,'Endodontics','#20 hand files',126.8151,'NO',1081);

Inserted records for another 497 records with the similar insert script.

	PRODUCT_ID	PRODUCT_TYPE	PRODUCT_NAME	PRODUCT_PRICE	PROD_AVAILABILITY	VENDOR_ID
1	214	Endodontics	#6 hand files	9746	NO	1105
2	215	Endodontics	#8 hand files	4976	NO	1105
3	216	Endodontics	#10 hand files	9429	NO	1129
4	217	Endodontics	#15 hand files	4445	NO	1085
5	218	Endodontics	#20 hand files	7210	NO	1081
6	219	Endodontics	#25 hand files	4823	YES	1102
7	220	Endodontics	#30 hand files	9160	NO	1006
8	221	Endodontics	#35 hand files	9435	YES	1064
9	222	Endodontics	#40 hand files	915	NO	1113
10	223	Endodontics	Barbed Broach - Diff Sizes	2940	YES	1124
11	224	Endodontics	Rotary System files	2333	YES	1005
12	225	Endodontics	Gutta Percha	2460	YES	1082
13	226	Endodontics	Root canal Cement	6331	NO	1134
14	227	Endodontics	Cavit	1420	NO	1014
15	228	Endodontics	Peper points	118	NO	1021
16	229	Endodontics	calcium hydroxide powder	5965	YES	1078
17	230	Impression Trays and Materials	Triple tray	8875	NO	1026
18	231	Impression Trays and Materials	Upper full arch alginate tra...	2400	YES	1093

ORDERS

```
create or replace PROCEDURE proc_data_generation
IS

BEGIN
    dbms_output.put_line('This is begin section');
    FOR I IN 1..200
        LOOP
            --insert records into orders table
            INSERT INTO ORDERS (ORDER_ID
                                ,ORDER_DATE
                                ,DELIVERY_DUE_DATE
                                ,PAYMENT_MODE
                                ,TOTAL_PRICE
                                ,USER_ID)
                        VALUES (ORDER_SEQ.NEXTVAL
                                ,SYSDATE
                                ,SYSDATE
                                ,'ONLINE'
                                ,0
                                ,5);
            COMMIT;
            dbms_output.put_line('RECORD INSERTED SUCCESSFULY');

            --manipulating records in orders table
            UPDATE ORDERS
            SET DELIVERY_DUE_DATE = '01-JUN-21';

            UPDATE ORDERS
            SET TOTAL_PRICE = ROUND(DBMS_RANDOM.VALUE(1,80000));

            COMMIT;
        END LOOP;

EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('THIS IS EXCEPTION SECTION');
END;
```

	ORDER_ID	ORDER_DATE	DELIVERY_DUE_DATE	PAYMENT_MODE	TOTAL_PRICE	USER_ID
1	6	05-FEB-21	01-JUN-21	ONLINE	49500	107
2	55	23-DEC-20	15-DEC-21	CASH	76010	107
3	9	21-DEC-20	01-JUN-21	ONLINE	73570	107
4	184	23-NOV-20	01-JUN-21	ONLINE	1011	108
5	4	22-NOV-20	01-JUN-21	ONLINE	9272	107
6	201	16-MAY-20	01-JUN-21	ONLINE	18260	108
7	198	14-MAY-20	01-JUN-21	ONLINE	59730	108
8	199	31-JAN-20	01-JUN-21	ONLINE	61490	108
9	7	12-DEC-19	01-JUN-21	ONLINE	15540	107
10	145	11-NOV-19	30-AUG-21	ONLINE	8675	108
11	21	16-OCT-19	01-JUN-21	ONLINE	47180	107
12	186	23-SEP-19	01-JUN-21	ONLINE	66330	108
13	29	26-FEB-19	01-JUN-21	ONLINE	5698	107
14	180	29-DEC-18	01-JUN-21	ONLINE	62830	108
15	134	23-DEC-18	30-AUG-21	ONLINE	66780	108
16	43	21-AUG-18	15-DEC-21	ONLINE	63160	107
17	162	03-AUG-18	01-JUN-21	CASH	73320	108
18	49	01-AUG-18	15-DEC-21	ONLINE	33820	107
19	187	30-MAY-18	01-JUN-21	ONLINE	31090	108
20	90	23-FEB-18	01-JUN-21	CASH	51520	107

ORDER_DETAILS

```
create or replace PROCEDURE proc_data_gen_dtls
IS

cursor cur_rec is
select order_id from orders order by order_id asc;

BEGIN
    dbms_output.put_line('This is begin section');
    FOR rec_cur IN cur_rec
    LOOP
        INSERT INTO order_details ( ORDER_DTL_ID
                                   ,QUANTITY
                                   ,PRICE
                                   ,ORDER_ID
                                   ,PRODUCT_ID)
                           values (ORDER_DTL_SEQ.nextval
                                   ,round(dbms_random.value(1,100))
                                   ,round(dbms_random.value(1,10000))
                                   ,rec_cur.order_id
                                   ,0 );
        COMMIT;
    END LOOP;
    dbms_output.put_line('RECORD INSERTED SUCCESSFULY');

EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('THIS IS EXCEPTION SECTION');
END proc_data_gen_dtls;
```

UPDATE order_details SET product_id = 341 WHERE order_dtl_id =1;
UPDATE order_details SET product_id = 342 WHERE order_dtl_id =2;
UPDATE order_details SET product_id = 343 WHERE order_dtl_id =3;
UPDATE order_details SET product_id = 344 WHERE order_dtl_id =4;
UPDATE order_details SET product_id = 345 WHERE order_dtl_id =5;
UPDATE order_details SET product_id = 346 WHERE order_dtl_id =6;

Updated records for 200 order detail id's with similar script

	ORDER_DTL_ID	QUANTITY	PRICE	ORDER_ID	PRODUCT_ID
1	1	97	3083	2	390
2	2	82	5430	3	177
3	3	93	2988	4	346
4	4	88	2487	5	160
5	5	47	2908	6	350
6	6	85	5484	7	351
7	7	28	8457	8	355
8	8	33	7610	9	337
9	9	4	4256	10	329
10	10	21	7403	11	350
11	11	77	2132	12	390
12	12	94	6741	13	177
13	13	18	2609	14	346
14	14	44	3210	15	160
15	15	78	5940	16	350
16	16	32	5117	17	351
17	17	58	6755	18	355
18	18	98	8660	19	337

USER_ROLE_MAPPING

```
create or replace PROCEDURE PROC_USER_ROLE_MAP_INSERT AS

CURSOR cur_user IS
SELECT user_id FROM user_master
WHERE user_id <> 100
ORDER BY user_id ASC;

CURSOR cur_patient IS
SELECT patient_id FROM patient_records
ORDER BY patient_id ASC;

CURSOR cur_vendor IS
SELECT vendor_id FROM vendor_details
ORDER BY vendor_id ASC;

BEGIN
  FOR rec_user IN cur_user
  LOOP
    INSERT INTO user_role_mapping (role_id,user_id) VALUES (2,rec_user.user_id);
  END LOOP;
  COMMIT;
  dbms_output.put_line('Record inserted successfully for Users');

  FOR rec_pat IN cur_patient
  LOOP
    INSERT INTO user_role_mapping (role_id,user_id) VALUES (3,rec_pat.patient_id);
  END LOOP;
  COMMIT;
  dbms_output.put_line('Record inserted successfully for Patients');

  FOR rec_vendor IN cur_vendor
  LOOP
    INSERT INTO user_role_mapping (role_id,user_id) VALUES (4,rec_vendor.vendor_id);
  END LOOP;
  COMMIT;
  dbms_output.put_line('Record inserted successfully for Vendors');

EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('THIS IS EXCEPTION SECTION');
END PROC_USER_ROLE_MAP_INSERT;
```

	ROLE_ID	USER_ID
19	3	1002
20	3	1003
21	3	1004
22	3	1005
23	3	1006
24	3	1007
25	3	1008
26	3	1009
27	3	1010
28	3	1011
29	3	1012
30	3	1013
31	3	1014
32	3	1015
33	3	1016
34	3	1017
35	3	1018
36	3	1019

QUERY WRITING & DATABASE PROGRAMMING

QUERIES

QUERY 1 – Display the list of patients (provide ID and name) that have appointments who live on abbey street. Order the results by patient ID.

```
SELECT
  DISTINCT(patient_id)
  , pfname, plname
  , p_address
FROM
   patient_records
INNER JOIN appointment_dtls
   USING(patient_id)
WHERE p_address LIKE '%ABBEY STREET%'
ORDER BY patient_id ;
```

OUTPUT :

	PATIENT_ID	PFNAME	PLNAME	P_ADDRESS
1	1093	Robert	Russell	32 ABBEY STREET
2	1099	Penelope	Barnes	32 ABBEY STREET
3	2050	Adelina	Goldstein	32 ABBEY STREET
4	2202	Ivanna	Courtney	32 ABBEY STREET

QUERY 2 – Display the total revenue of all the appointments that had been scheduled for Dr. Katnis Everdeen.

```
SELECT
  SUM(bill_amount) as TOTAL_REVENUE
  ,user_id
FROM
   billing_details
INNER JOIN appointment_dtls
   USING(appointment_id)
WHERE user_id= 112
AND bill_date > add_months(bill_date,-12)
GROUP BY user_id;
```

OUTPUT :

	TOTAL_REVENUE	USER_ID
1	326691	112

QUERY 3 – Display List of patients, dental history, treatments who have taken appointment at least thrice.

```
SELECT
   COUNT(a.appointment_id) as APPT_COUNT,
   a.patient_id,
   p.pfname,
   p.plname,
   p.dental_history,
   t.treatment_name
FROM
   appointment_dtls a
INNER JOIN patient_records p
   ON(a.patient_id= p.patient_id)
INNER JOIN patient_treatment_mapping ptm
   ON (a.appointment_id= ptm.appointment_id)
INNER JOIN treatments t
   ON(ptm.treatment_id=t.treatement_id)
GROUP BY
   a.patient_id,
   p.pfname,
   p.plname,
   p.dental_history,
   t.treatment_name
```

```
HAVING COUNT(a.appointment_id)>=3
ORDER BY
   COUNT(a.appointment_id) DESC;
```

OUTPUT :

	APPT_COUNT	PATIENT_ID	PFNAME	PLNAME	DENTAL_HISTORY	TREATMENT_NAME
1	3	3146	Abbie	Nagy	The Patient visited for a Skilled Nursing Facility	facial pain
2	3	3016	Honesty	Whitt	The Patient visited for a Durable Medical Equipment	facial pain
3	3	4831	Rosalinda	Sonowal	The Patient visited for a Laboratory Outpatient and Professional Services	Dental sealants 6-11 years
4	3	1897	Emmalyn	Donaldson	The Patient visited for a Outpatient Rehabilitation Services	Dental caries (tooth decay)
5	3	2789	Vincenzo	Lovett	The Patient visited for a Substance Abuse Disorder Inpatient Services	Dental caries (tooth decay)

QUERY 4 - Display top 10 products that have been ordered by the clinicians along with their vendor details.

```
SELECT
   COUNT(od.order_dtl_id) product_count
   ,od.product_id
   ,pr.product_name
   ,vd.vendor_name
   ,vd.vendor_type
FROM
   order_details od
   , products pr
   ,vendor_details vd
WHERE od.product_id = pr.product_id
AND pr.vendor_id = vd.vendor_id
GROUP BY
   od.product_id
   ,pr.product_name
   ,vd.vendor_name
   ,vd.vendor_type
ORDER BY
   COUNT(od.order_dtl_id) DESC
FETCH FIRST 10 ROWS ONLY;
```

OUTPUT :

	PRODUCT_COUNT	PRODUCT_ID	PRODUCT_NAME	VENDOR_NAME	VENDOR_TYPE
1	25	329	Masks	You First Services	faceshield provider
2	20	351	Paper towel roll	VOCO America Inc	dental equipments
3	15	346	Cotton Rolls	US Dental Supplies	medicine provider
4	14	355	Saliva Ejectors	Ascentcare Dental Products	dental equipment
5	13	337	Sterilization pouches	VisionAir	faceshield provider
6	11	390	30 gauge short needles	e-Dentist Supply	medicine provider
7	10	246	Plaster knife	Dental District	faceshield provider
8	10	350	Ice-packs	Dentazon	faceshield provider
9	10	177	Crown Splitter - #N134	ALE Innovations	faceshield provider
10	8	402	stationary	Henry Schein	digital technology

QUERY 5 - Display the total cost of treatment done on patients till date.

```
SELECT
   qry1.treatment_name
   ,qry1.treatment_category
   ,qry1.cnt*qry1.coi_member total_cost_member
   ,qry1.cnt*qry1.coi_non_member total_cost_nonmember
FROM
   (
      SELECT
         COUNT(ptm.appointment_id) cnt
         ,ptm.treatment_id
         ,tt.treatment_name
         ,tt.treatment_category
         ,tt.coi_member
         ,tt.coi_non_member
      FROM
         patient_treatment_mapping ptm
         ,treatments tt
      WHERE ptm.treatment_id = tt.treatement_id
      GROUP BY
         ptm.treatment_id
         ,tt.treatment_name
         ,tt.treatment_category
         ,tt.coi_member
         ,tt.coi_non_member
```

```
    ORDER BY
        COUNT(ptm.appointment_id) DESC
) qry1

ORDER BY
    total_cost_member DESC
    ,total_cost_nonmember DESC;
```

OUTPUT :

	TREATMENT_NAME	TREATMENT_CATEGORY	TOTAL_COST_MEMBER	TOTAL_COST_NONMEMBER
1	Periodontal disease	Severe	1670214	1876294
2	Oral Cancer	1st Stage	1572258	1683964
3	Oral Cancer	3rd Stage	1565823	2362834
4	Oral Cancer	2nd Stage	1495819	1801202
5	facial pain	Extremely High Pain	1414446	1638732
6	Periodontal disease	Moderate	1373336	1583256
7	facial pain	Low Pain	1259995	1459790
8	facial pain	High Pain	1211930	1771060
9	Dental caries (tooth decay)	Dental Filling	1053527	1175988
10	Craniofacial defects	High	1043700	1220975
11	National Health and Nutrition Examination	(null)	1031184	1413456
12	Tooth loss	(null)	952200	1167000
13	Water Fluoridation	(null)	930584	1088114
14	Craniofacial defects	Average	865792	1046106
15	Dental caries (tooth decay)	Carries Removal	856044	995536
16	Dental sealants 6-11 years	(null)	817908	994084
17	Dental caries (tooth decay) 2-11 years	(null)	789660	997380
18	Dental sealants	(null)	761634	929385
19	Craniofacial defects	Low	756564	1161177
20	Dental caries (tooth decay)	Tooth Extraction	393843	602292

QUERY 6 – Display the peak hours of the clinic based on patients visit for new and followup appointments.

```
SELECT
   COUNT(appointment_id) CNT
   ,appt_time
FROM
    appointment_dtls
WHERE appt_status IN ('NEW','FOLLOWUP')
GROUP BY
    appt_time
ORDER BY
    COUNT(appointment_id) DESC
```

OUTPUT :

	CNT	APPT_TIME
1	442	9:00:00 AM
2	440	4:00:00 PM
3	440	3:00:00 PM
4	439	12:00:00 PM
5	439	11:00:00 AM
6	439	2:00:00 PM
7	438	10:00:00 AM
8	438	1:00:00 PM
9	406	7:00:00 PM
10	405	8:00:00 PM
11	405	6:00:00 PM
12	404	5:00:00 PM
13	204	9:00:00 PM

QUERY 7 - Display the treatment details having cost >6000 USD

```
SELECT
   treatment_name,
   description details,
   coi_member cost_mem,
   coi_non_member cost_nonmem
FROM
   Treatments
WHERE coi_member > 6000
AND coi_non_member > 6000
ORDER BY
   coi_member,
   coi_non_member;
```

OUTPUT :

	TREATMENT_NAME	DETAILS	COST_MEM	COST_NONMEM
1	Dental caries (tooth ...	Dental caries (tooth decay) remains the most prevalent chroni...	6977	7788
2	facial pain	The most common cause of facial pain is a group of conditions...	7129	10418
3	facial pain	The most common cause of facial pain is a group of conditions...	8129	9418
4	facial pain	The most common cause of facial pain is a group of conditions...	8129	9418
5	Periodontal disease	Overall, the prevalence of both moderate and severe periodont...	8374	9654
6	Oral Cancer	Approximately 49,700 Americans are diagnosed each year with t...	8851	10658
7	Oral Cancer	Approximately 49,700 Americans are diagnosed each year with t...	9051	13658
8	Oral Cancer	Approximately 49,700 Americans are diagnosed each year with t...	9951	10658
9	Periodontal disease	Overall, the prevalence of both moderate and severe periodont...	10374	11654
10	Veneers	Veneers are strong, thin pieces of porcelain that are bonded ...	14962	15996
11	Veneers	Veneers are strong, thin pieces of porcelain that are bonded ...	16962	20996
12	Veneers	Veneers are strong, thin pieces of porcelain that are bonded ...	20962	25996

QUERY 8 – Display which drug is prescribed the most by a dentist.

```
SELECT
   treatment_id,
   med_name,
   COUNT(med_name) AS Drug_Count
FROM
   patient_treatment_mapping
GROUP BY
   treatment_id,
   med_name
ORDER BY
   COUNT(med_name) DESC
FETCH FIRST 1 ROWS ONLY;
```

OUTPUT :

	TREATMENT_ID	MED_NAME	DRUG_COUNT
1	19	Antoxines multivit 100mg capsule	180

QUERY 9 – Display the treatment name, category, a textual label defining how costly the treatment is for each treatment based on following conditions.

a) If the cost of insured treatment is less than or equal to 5000 USD label it as "Inexpensive Treatment".

b) If the cost of insured treatment is greater than 5000 USD and less than or equal to10000 USD label it as "Moderately Expensive Treatment"

c) If the cost of insured treatment is greater than 10000 USD label it as "Highly Expensive Treatment".

```
SELECT
   treatment_name,
   treatment_category,
   coi_member cost,
   CASE
      WHEN coi_member <= 5000 THEN 'Inexpensive Treatment'
      WHEN coi_member > 5000 AND coi_member <= 10000 THEN 'Moderately Expensive Treatment'
      ELSE 'Highly Expensive Treatment'
   END AS COST_LABEL
FROM
   treatments;
```

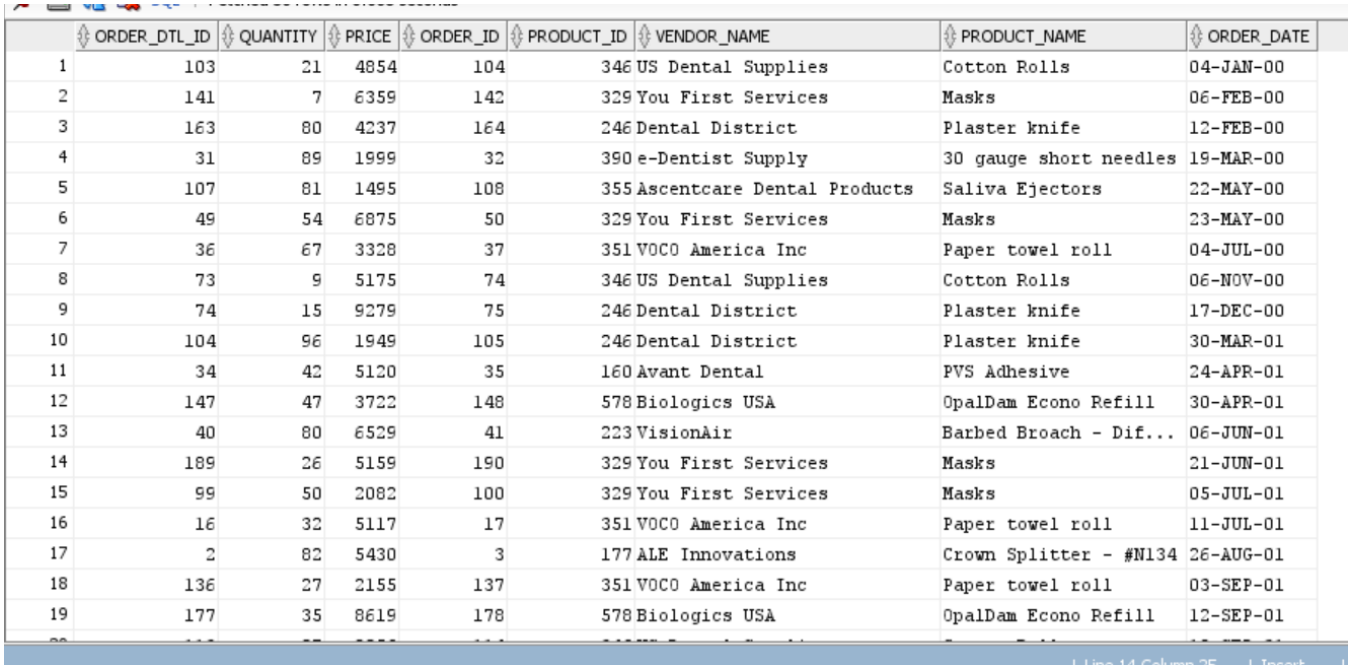
OUTPUT :

	TREATMENT_NAME	TREATMENT_CATEGORY	COST	COST_LABEL
1	Craniofacial defects	Low	3764	Inexpensive Treatment
2	Craniofacial defects	Average	4864	Inexpensive Treatment
3	Craniofacial defects	High	5964	Moderately Expensive Treatment
4	Dental caries (tooth decay)	Carries Removal	4977	Inexpensive Treatment
5	Dental caries (tooth decay)	Tooth Extraction	2477	Inexpensive Treatment
6	Dental caries (tooth decay)	Dental Filling	6977	Moderately Expensive Treatment
7	Dental sealants	(null)	4454	Inexpensive Treatment
8	facial pain	Low Pain	8129	Moderately Expensive Treatment
9	facial pain	High Pain	7129	Moderately Expensive Treatment
10	facial pain	Extremely High Pain	8129	Moderately Expensive Treatment
11	Oral Cancer	1st Stage	9951	Moderately Expensive Treatment
12	Oral Cancer	2nd Stage	8851	Moderately Expensive Treatment
13	Oral Cancer	3rd Stage	9051	Moderately Expensive Treatment
14	Periodontal disease	Moderate	8374	Moderately Expensive Treatment
15	Periodontal disease	Severe	10374	Highly Expensive Treatment
16	Tooth loss	(null)	4761	Inexpensive Treatment
17	National Health and Nutrition Examination	(null)	5859	Moderately Expensive Treatment
18	Water Fluoridation	(null)	5228	Moderately Expensive Treatment

QUERY 10 – In order to know the monthly expense of the clinic Display the details of orders made every month along with the products and their respective vendors.

```
SELECT
   odtl.order_dtl_id,
   odtl.quantity,
   odtl.price,
   odtl.order_id,
   ord.order_date,
   vd.vendor_name,
   p.product_name
FROM
   Order_details odtl
   INNER JOIN orders ord
      ON (odtl.order_id = ord.order_id)
   INNER JOIN products p
      ON (odtl.product_id =p.product_id)
   INNER JOIN vendor_details vd
      ON (p.vendor_id =vd.vendor_id)
ORDER BY
   ord.order_date;
```

OUTPUT :

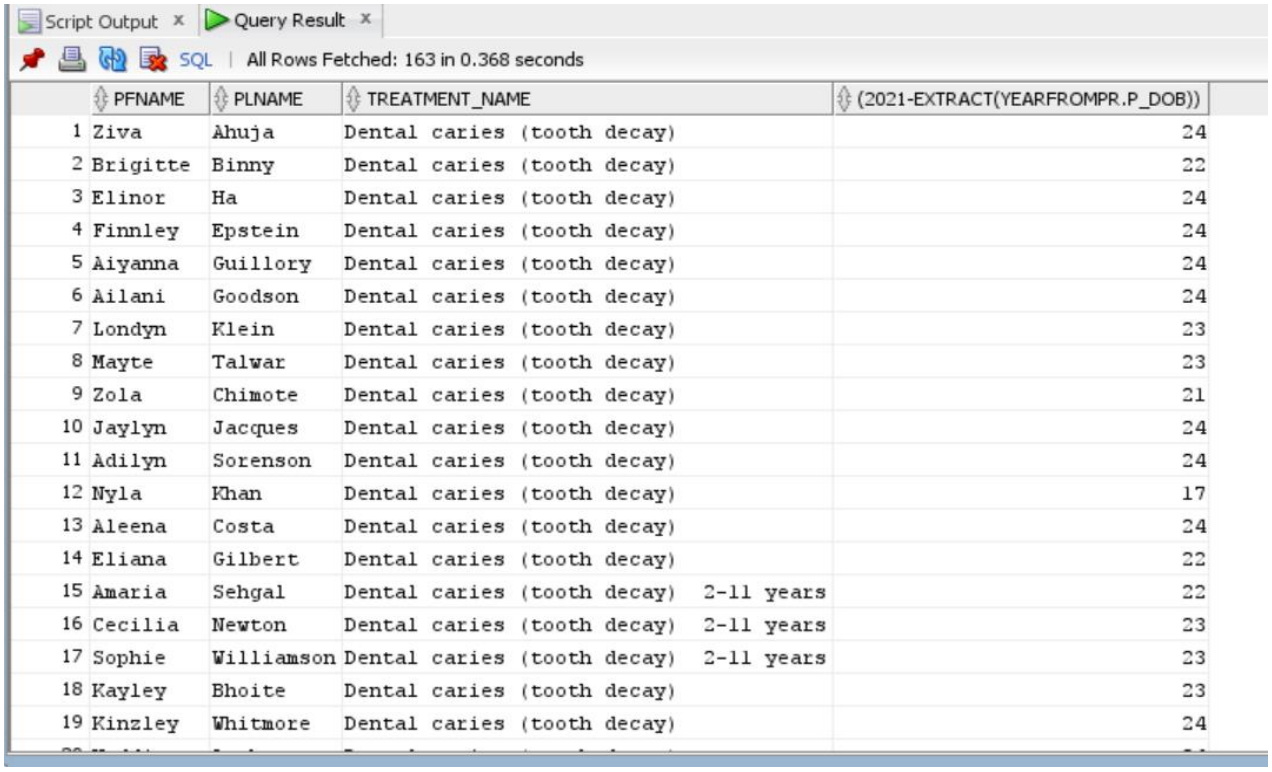


	ORDER_DTL_ID	QUANTITY	PRICE	ORDER_ID	PRODUCT_ID	VENDOR_NAME	PRODUCT_NAME	ORDER_DATE
1	103	21	4854	104	346	US Dental Supplies	Cotton Rolls	04-JAN-00
2	141	7	6359	142	329	You First Services	Masks	06-FEB-00
3	163	80	4237	164	246	Dental District	Plaster knife	12-FEB-00
4	31	89	1999	32	390	e-Dentist Supply	30 gauge short needles	19-MAR-00
5	107	81	1495	108	355	Ascentcare Dental Products	Saliva Ejectors	22-MAY-00
6	49	54	6875	50	329	You First Services	Masks	23-MAY-00
7	36	67	3328	37	351	VOCO America Inc	Paper towel roll	04-JUL-00
8	73	9	5175	74	346	US Dental Supplies	Cotton Rolls	06-NOV-00
9	74	15	9279	75	246	Dental District	Plaster knife	17-DEC-00
10	104	96	1949	105	246	Dental District	Plaster knife	30-MAR-01
11	34	42	5120	35	160	Avant Dental	PVS Adhesive	24-APR-01
12	147	47	3722	148	578	Biologics USA	OpalDam Econo Refill	30-APR-01
13	40	80	6529	41	223	VisionAir	Barbed Broach - Dif...	06-JUN-01
14	189	26	5159	190	329	You First Services	Masks	21-JUN-01
15	99	50	2082	100	329	You First Services	Masks	05-JUL-01
16	16	32	5117	17	351	VOCO America Inc	Paper towel roll	11-JUL-01
17	2	82	5430	3	177	ALE Innovations	Crown Splitter - #N134	26-AUG-01
18	136	27	2155	137	351	VOCO America Inc	Paper towel roll	03-SEP-01
19	177	35	8619	178	578	Biologics USA	OpalDam Econo Refill	12-SEP-01

QUERY 11 – Display the list of females below age of 25 that have undergone tooth decay treatments.

```
SELECT
   pr.pfname,
   pr.plname,
   t.treatment_name,
   (2021-EXTRACT(YEAR FROM pr.p_dob))
FROM
   patient_records pr
   INNER JOIN patient_treatment_mapping ptm
      ON (pr.patient_id=ptm.pt_id)
   INNER JOIN treatments t
      ON (ptm.treatment_id=t.treatement_id)
WHERE pr.p_gender='F'
AND t.treatment_name LIKE '%tooth decay%'
AND (2021-EXTRACT(YEAR FROM pr.p_dob)<25)
GROUP BY
   pr.pfname,
   pr.plname,
   t.treatment_name,
   pr.p_dob;
```

OUTPUT :



Script Output | Query Result

SQL | All Rows Fetched: 163 in 0.368 seconds

	PFNAME	PLNAME	TREATMENT_NAME	(2021-EXTRACT(YEARFROMPR.P_DOB))
1	Ziva	Ahuja	Dental caries (tooth decay)	24
2	Brigitte	Binny	Dental caries (tooth decay)	22
3	Elinor	Ha	Dental caries (tooth decay)	24
4	Finnley	Epstein	Dental caries (tooth decay)	24
5	Aiyanna	Guillory	Dental caries (tooth decay)	24
6	Ailani	Goodson	Dental caries (tooth decay)	24
7	Londyn	Klein	Dental caries (tooth decay)	23
8	Mayte	Talwar	Dental caries (tooth decay)	23
9	Zola	Chimote	Dental caries (tooth decay)	21
10	Jaylyn	Jacques	Dental caries (tooth decay)	24
11	Adilyn	Sorenson	Dental caries (tooth decay)	24
12	Nyla	Khan	Dental caries (tooth decay)	17
13	Aleena	Costa	Dental caries (tooth decay)	24
14	Eliana	Gilbert	Dental caries (tooth decay)	22
15	Amaria	Sehgal	Dental caries (tooth decay) 2-11 years	22
16	Cecilia	Newton	Dental caries (tooth decay) 2-11 years	23
17	Sophie	Williamson	Dental caries (tooth decay) 2-11 years	23
18	Kayley	Bhoite	Dental caries (tooth decay)	23
19	Kinzley	Whitmore	Dental caries (tooth decay)	24

QUERY 12 – Display names of all the Active dentist and count of appointment for each of them per month.

```
SELECT
   q1.ufname,
   q1.ulname,
   q2.cnt_appt_id,
   q2.Month_of_Appt,
   q2.Year_of_Appt
FROM
   (SELECT
      um.ufname,
      um.ulname
    FROM
      user_master um
    WHERE um.user_type_id=2
    AND um.user_status='ACTIVE'
   )q1,
   (SELECT
      COUNT (ad.appointment_id)cnt_appt_id,
      EXTRACT(MONTH FROM ad.appt_date) Month_of_Appt,
      EXTRACT(YEAR FROM ad.appt_date) Year_of_Appt,
```

```
    ad.user_id
  FROM appointment_dtls ad
  GROUP BY
    user_id,
    EXTRACT(MONTH FROM ad.appt_date),
    EXTRACT(YEAR FROM ad.appt_date)
  )q2
WHERE q1.user_id = q2.user_id
ORDER BY
  q1.user_id;
```

	UFNAME	ULNAME	CNT_APPT_ID	MONTH_OF_APPT	YEAR_OF_APPT
5	James	Bond	689	10	2016
6	James	Bond	196	11	2016
7	James	Bond	139	12	2016
8	James	Bond	93	1	2017
9	James	Bond	74	2	2017
10	James	Bond	31	3	2017
11	James	Bond	30	4	2017
12	James	Bond	35	5	2017
13	James	Bond	60	6	2017
14	James	Bond	62	7	2017
15	James	Bond	62	8	2017
16	James	Bond	60	9	2017

QUERY 13 – Display the list of Patients who are not insured and have had a total charge greater than or equal to 55000 over the past month.

```
 SELECT
   pr.pfname,
   pr.plname,
   bd.bill_amount,
   EXTRACT(MONTH FROM bd.bill_date) Month_of_Billing
FROM
   patient_records pr
   INNER JOIN patient_treatment_mapping ptm
     ON (pr.patient_id= ptm.pt_id)
   INNER JOIN billing_details bd
     ON (ptm.appointment_id=bd.appointment_id)
WHERE pr.is_insured='N'
AND bd.bill_amount >55000
AND (( EXTRACT(MONTH FROM bd.bill_date)=EXTRACT(MONTH FROM SYSDATE)-1)
OR ( EXTRACT(MONTH FROM bd.bill_date)=EXTRACT(MONTH FROM SYSDATE)));
```

	PFNAME	PLNAME	BILL_AMOUNT	MONTH_OF_BILLING
1	Kyler	Underwood	86950	3

DATABASE PROGRAMMING

Procedure to send automated emails for Appointment Reminder

We have written a procedure to send an automated reminder email generated through system to patient two days before their appointment date using the UTL_SMTP and UTL_TCP packages of oracle. The packages use various functions to perform handshake and send mails. We have scheduled a job to send the emails every day at 9 AM which would pick all the required information and send emails.

```
create or replace PROCEDURE PROC_SEND_REMINDER_MAIL AS

v_from                  VARCHAR2(100) := 'dhms@systemmail.com';
v_subject               VARCHAR2(100) := 'Appointment Reminder Email';
v_smtp_host             VARCHAR2(10) := '10.0.0.127';
v_smtp_port             NUMBER := 25;
crlf                    VARCHAR2(10) := CHR(13)||CHR(10);
v_con                   UTL_SMTP.CONNECTION;
v_to                    VARCHAR2(100);
v_msg                   VARCHAR2(4000);

CURSOR cur_rec IS
SELECT
    ad.appt_date,
    ad.appt_time,
    pr.pfname||' '||pr.plname AS pname,
    pr.pemail_id
FROM
    appoINtment_Dtls ad
    , patient_records pr
WHERE appt_status = 'FOLLOW-UP'
AND ad.patient_id = pr.patient_id
AND UPPER(pr.patient_status) = 'ACTIVE'
AND ad.appt_date = SYSDATE + 2 ;
```

```
BEGIN

  v_con := UTL_SMTP.OPEN_CONNECTION(v_smtp_host,v_smtp_port);

  UTL_SMTP.HELO(v_con,v_smtp_host);
  UTL_SMTP.MAIL(v_con,v_from);

  FOR rec IN cur_rec
  LOOP
    v_to := rec.pemail_id;
    UTL_SMTP.RCPT(v_con,v_to);--to emails ids
  END LOOP;

  UTL_SMTP.OPEN_DATA(v_con);
  UTL_SMTP.WRITE_DATA(v_con,'Date: '||to_char(sysdate,'dd-mon-yyyy hh24:mi:ss')||crlf);
  UTL_SMTP.WRITE_DATA(v_con,'From: '||v_from||crlf);
  UTL_SMTP.WRITE_DATA(v_con,'Subject: '||v_subject||crlf);

  FOR rec IN cur_rec
  LOOP
    UTL_SMTP.WRITE_DATA(v_con,'To: '||v_to||crlf);
    v_msg := 'Dear'|| rec.pname|| ' This is a reminder email that You have an appintment on
    '|| rec.appt_date||' - '|| rec.appt_time|| ' Request you to get your insuarence
    and photo id card for your appointment' ;

  END LOOP;

  UTL_SMTP.WRITE_DATA(v_con,v_msg||crlf);
  UTL_SMTP.CLOSE_DATA(v_con);

  UTL_SMTP.QUIT(v_con);

END PROC_SEND_REMINDER_MAIL;
```

Job Script:

```
BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
   job_name         =>  'Appointment_reminder',
   program_name     =>  'PROC_SEND_REMINDER_MAIL',
   start_date       =>  '17-APR-2021 09:00:00 AM',
   repeat_interval  =>  'FREQ=DAILY',
   comments         =>  'Daily at 9am');
END;
/
```

PERFORMANCE TUNING

Performance tuning is the process of reducing the resource consumption or to reduce the elapsed time for an operation to complete. The goal is to improve the effective use of a particular resource. There are basically two types of tuning:

a) **Proactive Monitoring -** Usually occurs on a regularly scheduled interval, where several performance statistics are examined to identify whether the system behavior and resource usage has changed.

b) **Bottleneck Elimination -** The overused resource is the bottleneck in the system. There are several distinct phases in identifying the bottleneck and the potential fixes.

When a SQL statement is executed on an Oracle database, the query optimizer determines the most efficient execution plan after considering many factors related to the objects referenced and the conditions specified in the query. This determination is an important step in the processing of any SQL statement and can greatly affect execution time.

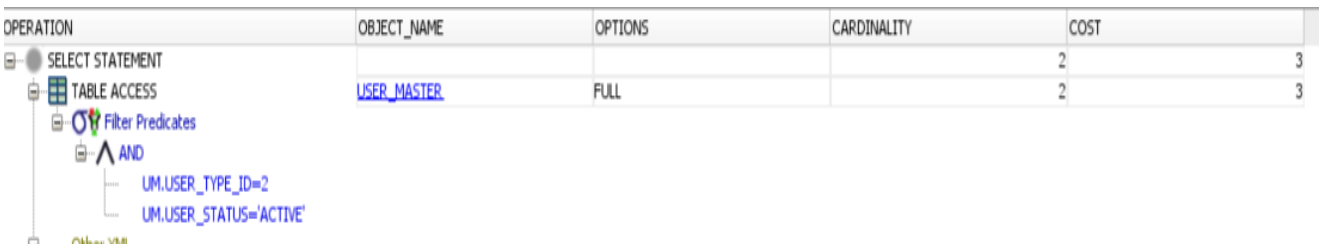
USING INDEXES

An index can sometimes speed data access. By creating an index on one or more columns of a table, we gain the ability in some cases to retrieve a small set of randomly distributed rows from the table. Indexes are one of many means of reducing disk I/O.

B-TREE INDEX - These indexes are the standard index type. They are excellent for primary key and highly-selective indexes. To check how B-tree index affects the performance, we created below index and compared results.

```
SELECT
   um.*
FROM
   user_master um
WHERE um.user_type_id=2
AND um.user_status='ACTIVE';
```

Without any index – The table has full access.



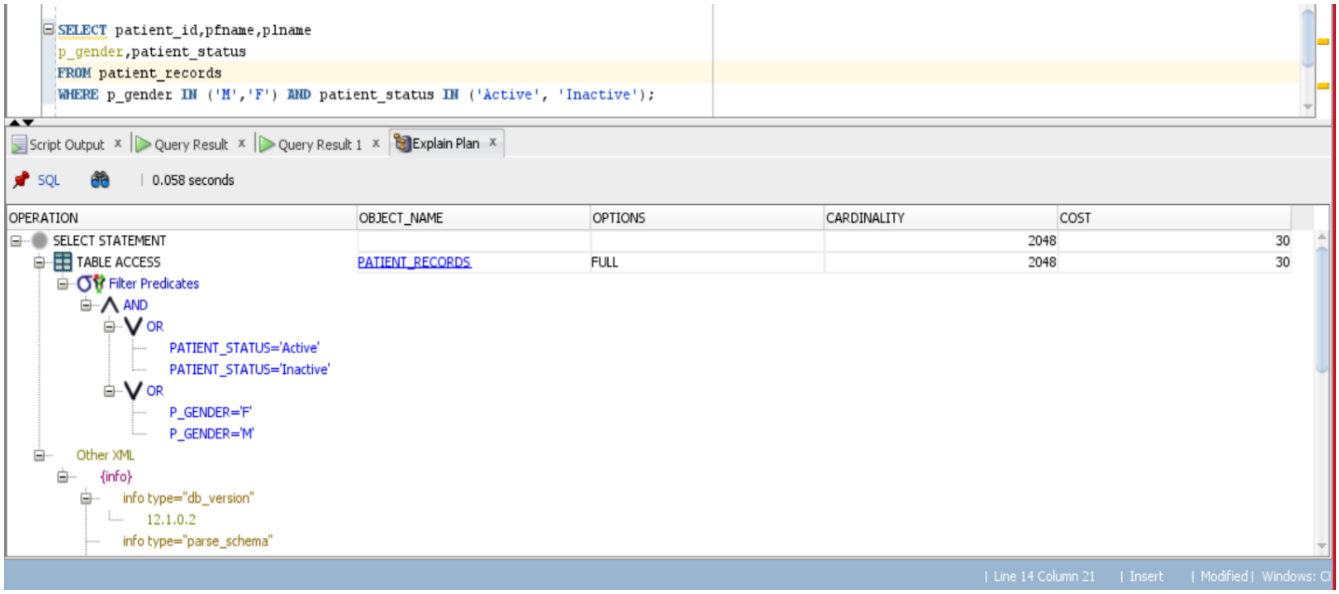
Index Script:
CREATE INDEX um_user_type_idx
ON user_master(user_type_id);

With index creation – The data is accessed using index which improved the cost and the full table access is removed.

BITMAP INDEX - In a bitmap index, an index entry uses a bitmap to point to multiple rows. To check how Bitmap index affects the performance, we created below index and compared results.

```
SELECT
   patient_id,
   pfname,
   plname,
   p_gender,
   patient_status
FROM
   patient_records
WHERE p_gender IN ('M','F')
AND patient_status IN ('Active');
```

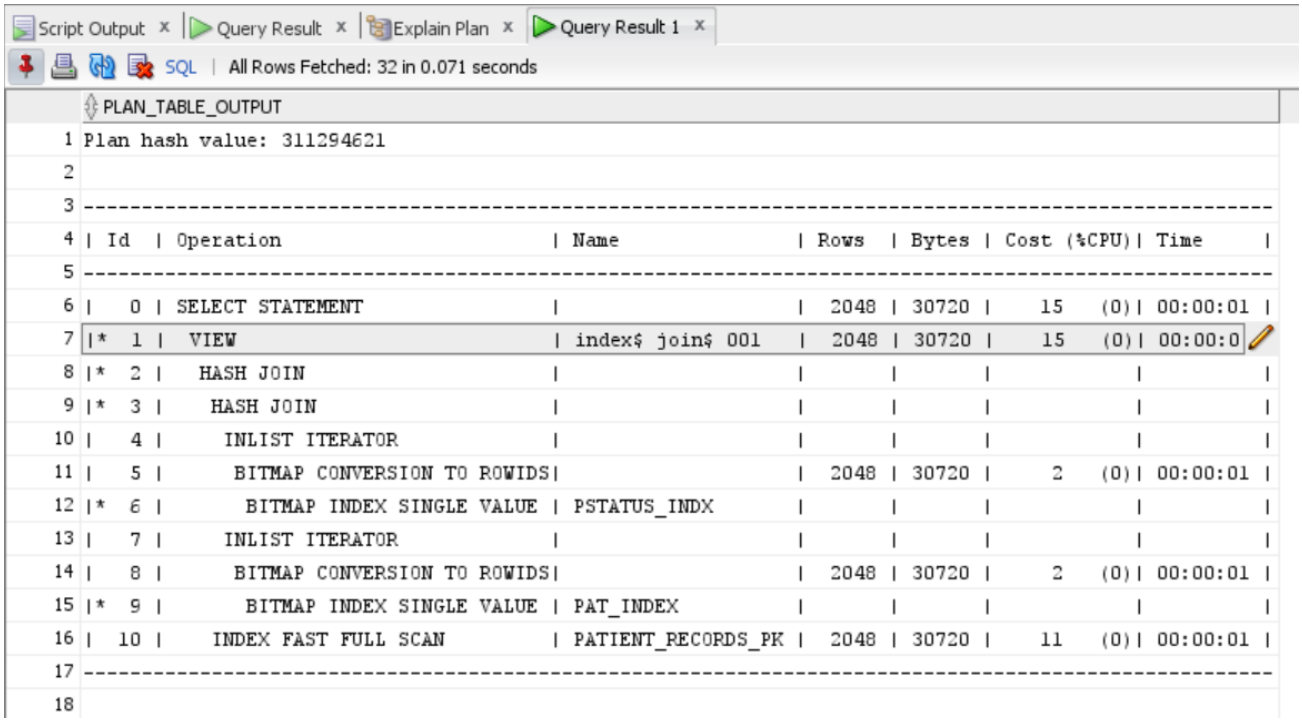
Without any index – The table has full access.



Index Script:
CREATE BITMAP INDEX pat_index
ON patient_records (p_gender);

CREATE BITMAP INDEX status_index
ON patient_records (patient_status);

With index creation – The data is accessed using index which improved the cost and the full table access is removed.

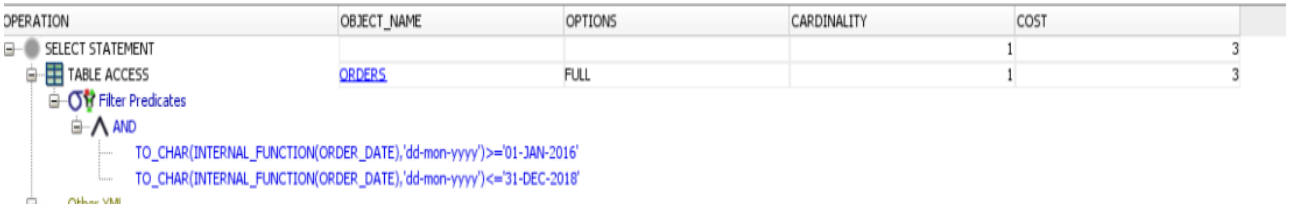


	PLAN_TABLE_OUTPUT
1	Plan hash value: 311294621
2	
3	----------
4	| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |
5	----------
6	| 0 | SELECT STATEMENT | | 2048 | 30720 | 15 (0)| 00:00:01 |
7	|* 1 | VIEW | index$ join$ 001 | 2048 | 30720 | 15 (0)| 00:00:0
8	|* 2 | HASH JOIN | | | | | |
9	|* 3 | HASH JOIN | | | | | |
10	| 4 | INLIST ITERATOR | | | | | |
11	| 5 | BITMAP CONVERSION TO ROWIDS| | 2048 | 30720 | 2 (0)| 00:00:01 |
12	|* 6 | BITMAP INDEX SINGLE VALUE | PSTATUS_INDX | | | | |
13	| 7 | INLIST ITERATOR | | | | | |
14	| 8 | BITMAP CONVERSION TO ROWIDS| | 2048 | 30720 | 2 (0)| 00:00:01 |
15	|* 9 | BITMAP INDEX SINGLE VALUE | PAT_INDEX | | | | |
16	| 10 | INDEX FAST FULL SCAN | PATIENT_RECORDS_PK | 2048 | 30720 | 11 (0)| 00:00:01 |
17	----------
18	

FUNCTION-BASED INDEX - This type of index includes columns that are either transformed by a function, such as the UPPER function, or included in an expression. B-tree or bitmap indexes can be function-based. To check how Funtion-based index affects the performance, we created below index and compared results.

```
SELECT
   ord.*
FROM
   orders ord
WHERE to_char(ord.order_date,'dd-mon-yyyy') BETWEEN '01-JAN-2016' AND '31-DEC-2018';
```

Without any index – The table has full access.



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	3
TABLE ACCESS	ORDERS	FULL	1	3
Filter Predicates				
AND				
TO_CHAR(INTERNAL_FUNCTION(ORDER_DATE),'dd-mon-yyyy')>='01-JAN-2016'				
TO_CHAR(INTERNAL_FUNCTION(ORDER_DATE),'dd-mon-yyyy')<='31-DEC-2018'				

Index Script:
CREATE INDEX order_date_idx
ON orders (to_char(order_date,'dd-mon-yyyy'));

With index creation – The data is accessed using index which improved the cost and the full table access is removed.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	2
TABLE ACCESS	ORDERS	BY INDEX ROWID BATCHED	1	2
INDEX	ORDER_DATE_IDX	RANGE SCAN	1	1
Access Predicates				
AND				

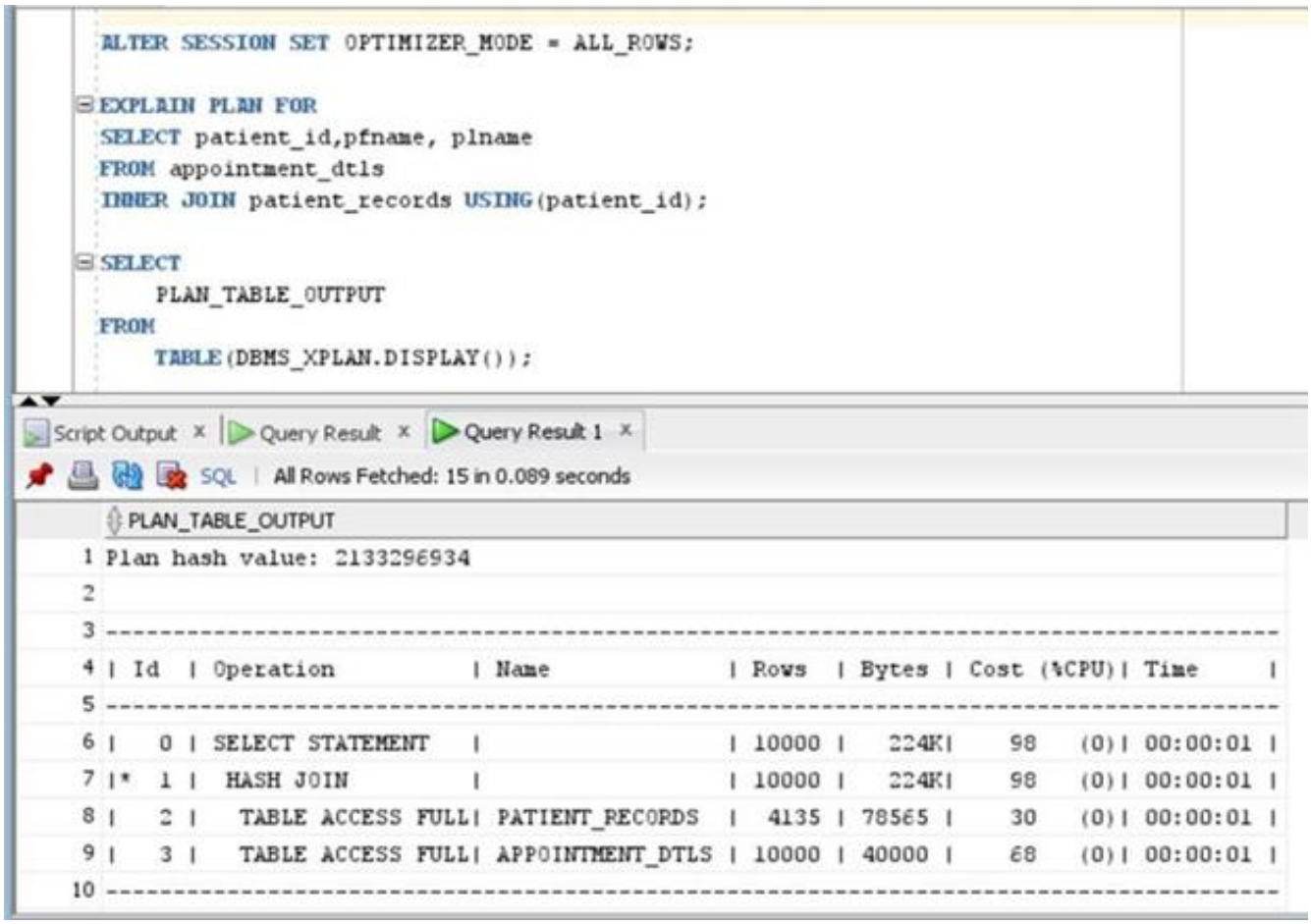
Below table shows the list of indexes created on tables

	INDEX_NAME	INDEX_TYPE	TABLE_NAME
1	ORDER_DATE_IDX	FUNCTION-BASED NORMAL	ORDERS
2	PAT_INDEX	BITMAP	PATIENT_RECORDS
3	PSTATUS_INDX	BITMAP	PATIENT_RECORDS
4	PR_COMP_IDX	NORMAL	PRODUCTS
5	PR_PRODAV_IDX	BITMAP	PRODUCTS
6	PR_PRODPRICE_IDX	BITMAP	PRODUCTS
7	PR_PRODTYPE_IDX	BITMAP	PRODUCTS
8	PR_VENID_IDX	NORMAL	PRODUCTS
9	TREAT_NAME_IDX	FUNCTION-BASED NORMAL	TREATMENTS
10	UM_CONT_IDX	BITMAP	USER_MASTER
11	UM_UEMAIL_IDX	BITMAP	USER_MASTER
12	UM_USER_TYPE_IDX	NORMAL	USER_MASTER
13	UM_USTATUS_IDX	BITMAP	USER_MASTER
14	IS_ACTIVE_IDX	BITMAP	VENDOR_DETAILS
15	VENDOR_TYPE_IDX	BITMAP	VENDOR_DETAILS

OPTIMIZER MODES

ALL_ROWS - It is the default optimizer mode; it gets all rows faster (generally forces index suppression). This is good for untuned, high-volume batch systems.

On executing the query, we see that the cost of query processing is 98.



```
ALTER SESSION SET OPTIMIZER_MODE = ALL_ROWS;

EXPLAIN PLAN FOR
SELECT patient_id,pfname, plname
FROM appointment_dtls
INNER JOIN patient_records USING(patient_id);

SELECT
    PLAN_TABLE_OUTPUT
FROM
    TABLE(DBMS_XPLAN.DISPLAY());
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 15 in 0.089 seconds

```
PLAN_TABLE_OUTPUT
1  Plan hash value: 2133296934
2
3  ------------------------------------------------------------------------------------
4  | Id  | Operation          | Name             | Rows  | Bytes | Cost (%CPU)| Time     |
5  ------------------------------------------------------------------------------------
6  |   0 | SELECT STATEMENT   |                  | 10000 |   224K|    98   (0)| 00:00:01 |
7  |*  1 |  HASH JOIN         |                  | 10000 |   224K|    98   (0)| 00:00:01 |
8  |   2 |   TABLE ACCESS FULL| PATIENT_RECORDS  |  4135 | 78565 |    30   (0)| 00:00:01 |
9  |   3 |   TABLE ACCESS FULL| APPOINTMENT_DTLS | 10000 | 40000 |    68   (0)| 00:00:01 |
10 ------------------------------------------------------------------------------------
```

FIRST_ROWS_10 - It gets the first 10 rows faster. This is good for applications that routinely display partial results to users such as paging data to a user in a web application.

On executing the same above query, we see that the cost of processing has reduced from 98 to 12.

```
PLAN_TABLE_OUTPUT
Plan hash value: 2401842682

-------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name              | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                   |    10 |   230 |    12   (0)| 00:00:01 |
|   1 |  NESTED LOOPS                |                   |    10 |   230 |    12   (0)| 00:00:01 |
|   2 |   NESTED LOOPS               |                   |    10 |   230 |    12   (0)| 00:00:01 |
|   3 |    TABLE ACCESS FULL         | APPOINTMENT_DTLS  |    10 |    40 |     2   (0)| 00:00:01 |
|* 4 |    INDEX UNIQUE SCAN         | PATIENT_RECORDS_PK |     1 |       |     0   (0)| 00:00:01 |
|   5 |   TABLE ACCESS BY INDEX ROWID| PATIENT_RECORDS   |     1 |    19 |     1   (0)| 00:00:01 |
```

PARALLEL EXECUTION

Parallel execution divides the task of executing an SQL statement into multiple small units, each of which is executed by a separate process. Parallel execution is designed to effectively use multiple CPUs and disks to answer queries quickly. When multiple users use parallel execution at the same time, it is easy to quickly exhaust available CPU, memory, and disk resources.

DEGREE OF PARALLELISM -The number of parallel execution servers associated with a single operation is known as the degree of parallelism.

WITHOUT PARALLELISM –

```
SELECT
        COUNT(appointment_id),
        patient_id,
        pfname,
        plname
FROM
        appointment_dtls
INNER JOIN patient_records
        USING(patient_id)
INNER JOIN patient_treatment_mapping ptm
        USING(appointment_id)
INNER JOIN treatments t
        ON(ptm.treatment_id=t.treatement_id)
GROUP BY
```

```
        patient_id,
        pfname,
        plname
HAVING COUNT(appointment_id)>=3
ORDER BY
        COUNT(appointment_id) DESC ;
```

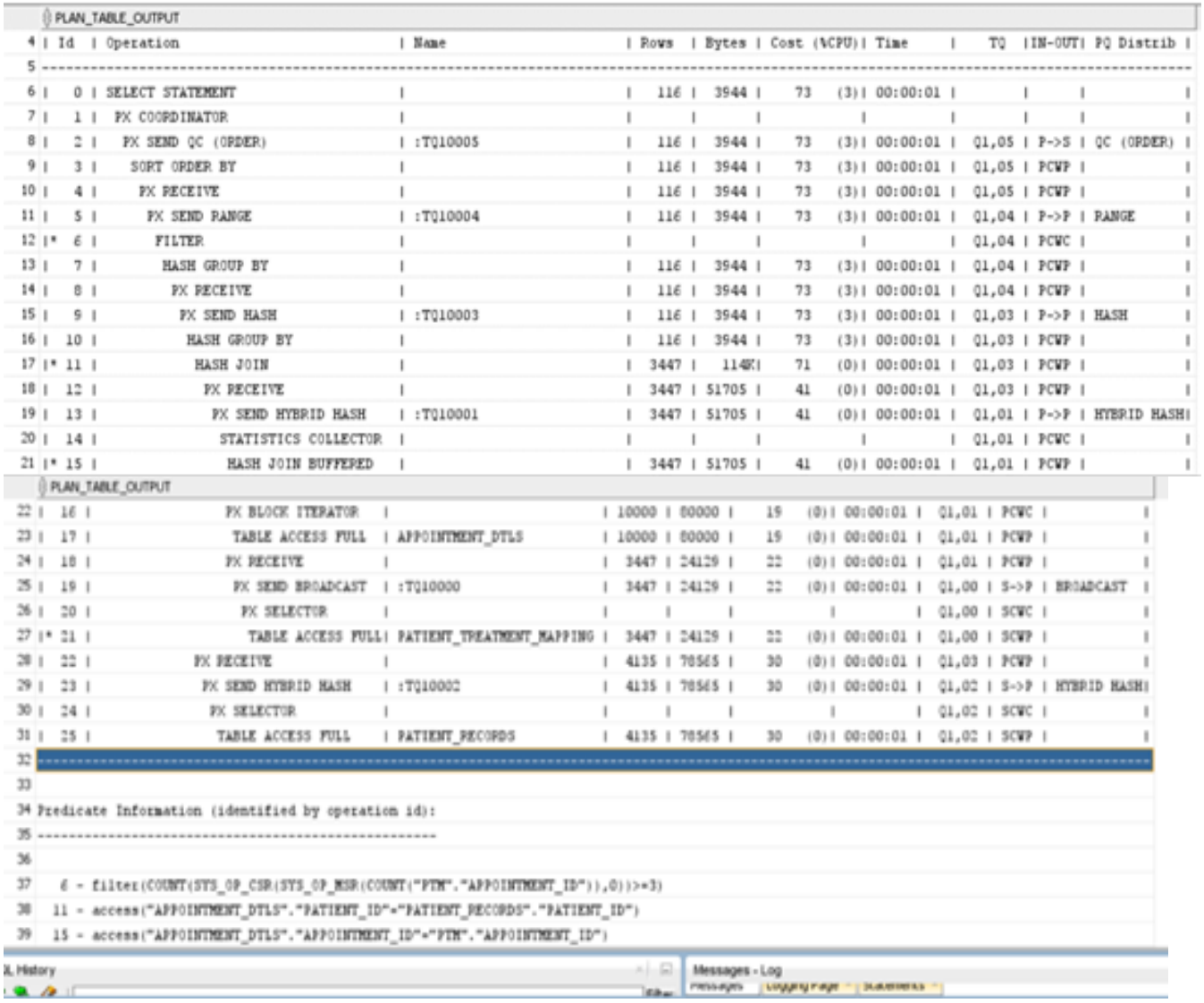
On executing the above query, we see that the cost of execution for non-parallel process is 122

```
PLAN_TABLE_OUTPUT
Plan hash value: 1913165403

-----------------------------------------------------------------------------------------------------
| Id  | Operation              | Name                       | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |                            |   116 |  3944 |   122   (2)| 00:00:01 |
|   1 |  SORT ORDER BY         |                            |   116 |  3944 |   122   (2)| 00:00:01 |
|*  2 |   FILTER               |                            |       |       |            |          |
|   3 |    HASH GROUP BY       |                            |   116 |  3944 |   122   (2)| 00:00:01 |
|*  4 |     HASH JOIN          |                            |  3447 |   114K|   120   (0)| 00:00:01 |
|*  5 |      HASH JOIN         |                            |  3447 | 51705 |    90   (0)| 00:00:01 |
|*  6 |       TABLE ACCESS FULL| PATIENT_TREATMENT_MAPPING  |  3447 | 24129 |    22   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL| APPOINTMENT_DTLS           | 10000 | 80000 |    68   (0)| 00:00:01 |
|   8 |      TABLE ACCESS FULL | PATIENT_RECORDS            |  4135 | 78565 |    30   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
```

PARALLELISM WITH DEGREE 4

ALTER TABLE appointment_dtls **PARALLEL** 4;

We see that after setting the degree of parallelism to 4, the cost of execution drastically drops to 73



PLAN_TABLE_OUTPUT

```
| Id  | Operation                     | Name                      | Rows  | Bytes | Cost (%CPU)| Time     |    TQ  |IN-OUT| PQ Distrib |
-------------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                           |   116 |  3944 |    73   (3)| 00:00:01 |        |      |            |
|   1 |  PX COORDINATOR               |                           |       |       |            |          |        |      |            |
|   2 |   PX SEND QC (ORDER)          | :TQ10005                  |   116 |  3944 |    73   (3)| 00:00:01 |  Q1,05 | P->S | QC (ORDER) |
|   3 |    SORT ORDER BY              |                           |   116 |  3944 |    73   (3)| 00:00:01 |  Q1,05 | PCWP |            |
|   4 |     PX RECEIVE                |                           |   116 |  3944 |    73   (3)| 00:00:01 |  Q1,05 | PCWP |            |
|   5 |      PX SEND RANGE            | :TQ10004                  |   116 |  3944 |    73   (3)| 00:00:01 |  Q1,04 | P->P | RANGE      |
|*  6 |       FILTER                  |                           |       |       |            |          |  Q1,04 | PCWC |            |
|   7 |        HASH GROUP BY          |                           |   116 |  3944 |    73   (3)| 00:00:01 |  Q1,04 | PCWP |            |
|   8 |         PX RECEIVE            |                           |   116 |  3944 |    73   (3)| 00:00:01 |  Q1,04 | PCWP |            |
|   9 |          PX SEND HASH         | :TQ10003                  |   116 |  3944 |    73   (3)| 00:00:01 |  Q1,03 | P->P | HASH       |
|  10 |           HASH GROUP BY       |                           |   116 |  3944 |    73   (3)| 00:00:01 |  Q1,03 | PCWP |            |
|* 11 |            HASH JOIN          |                           |  3447 |   114K|    71   (0)| 00:00:01 |  Q1,03 | PCWP |            |
|  12 |             PX RECEIVE        |                           |  3447 | 51705 |    41   (0)| 00:00:01 |  Q1,03 | PCWP |            |
|  13 |              PX SEND HYBRID HASH | :TQ10001                |  3447 | 51705 |    41   (0)| 00:00:01 |  Q1,01 | P->P | HYBRID HASH|
|  14 |               STATISTICS COLLECTOR |                       |       |       |            |          |  Q1,01 | PCWC |            |
|* 15 |                HASH JOIN BUFFERED |                       |  3447 | 51705 |    41   (0)| 00:00:01 |  Q1,01 | PCWP |            |
```

PLAN_TABLE_OUTPUT

```
|  16 |                 PX BLOCK ITERATOR  |                          | 10000 | 80000 |    19   (0)| 00:00:01 |  Q1,01 | PCWC |            |
|  17 |                  TABLE ACCESS FULL | APPOINTMENT_DTLS         | 10000 | 80000 |    19   (0)| 00:00:01 |  Q1,01 | PCWP |            |
|  18 |                 PX RECEIVE         |                          |  3447 | 24129 |    22   (0)| 00:00:01 |  Q1,01 | PCWP |            |
|  19 |                  PX SEND BROADCAST | :TQ10000                 |  3447 | 24129 |    22   (0)| 00:00:01 |  Q1,00 | S->P | BROADCAST  |
|  20 |                   PX SELECTOR       |                          |       |       |            |          |  Q1,00 | SCWC |            |
|* 21 |                    TABLE ACCESS FULL| PATIENT_TPEATMENT_MAPPING|  3447 | 24129 |    22   (0)| 00:00:01 |  Q1,00 | SCWP |            |
|  22 |               PX RECEIVE           |                          |  4135 | 78565 |    30   (0)| 00:00:01 |  Q1,03 | PCWP |            |
|  23 |                PX SEND HYBRID HASH | :TQ10002                 |  4135 | 78565 |    30   (0)| 00:00:01 |  Q1,02 | S->P | HYBRID HASH|
|  24 |                 PX SELECTOR        |                          |       |       |            |          |  Q1,02 | SCWC |            |
|  25 |                  TABLE ACCESS FULL | PATIENT_RECORDS          |  4135 | 78565 |    30   (0)| 00:00:01 |  Q1,02 | SCWP |            |
-------------------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   6 - filter(COUNT(SYS_OP_CSR(SYS_OP_MSR(COUNT("PTM"."APPOINTMENT_ID")),0))>=3)
  11 - access("APPOINTMENT_DTLS"."PATIENT_ID"="PATIENT_RECORDS"."PATIENT_ID")
  15 - access("APPOINTMENT_DTLS"."APPOINTMENT_ID"="PTM"."APPOINTMENT_ID")
```

History

Messages - Log

PARTITIONS

Partitioning of tables can improve the performance of data access substantially. To demonstrate this, we are creating partitions.

STEP 1 –

```
CREATE TABLE pat_treatment_map_part
( PT_ID NUMBER(30,0),
APPOINTMENT_ID NUMBER(30,0),
TREATMENT_ID NUMBER(10,0),
DOSAGE VARCHAR2(500),
MED_NAME VARCHAR2(500),
REMARKS VARCHAR2(500))
PARTITION BY RANGE (TREATMENT_ID)
(
PARTITION block1 VALUES LESS THAN (5),
PARTITION block2 VALUES LESS THAN (10),
PARTITION block3 VALUES LESS THAN (15),
PARTITION block4 VALUES LESS THAN (20),
PARTITION block5 VALUES LESS THAN (25),
PARTITION block6 VALUES LESS THAN (30),
PARTITION block7 VALUES LESS THAN (35),
PARTITION block8 VALUES LESS THAN (40)
);
```

STEP 2 -

```
INSERT INTO pat_treatment_map_part (PT_ID, APPOINTMENT_ID, TREATMENT_ID, DOSAGE,
MED_NAME, REMARKS)
SELECT pt_id, appointment_id, treatment_id, dosage, med_name, remarks FROM patient_treatment_mapping;
```

STEP 3 -

```
ANALYZE TABLE pat_treatment_map_part COMPUTE STATISTICS;

SELECT
  table_name,
  partition_name,
  num_rows,
  avg_row_len,
  last_analyzed
FROM user_tab_partitions
WHERE table_name = 'PAT_TREATMENT_MAP_PART';
```

	TABLE_NAME	PARTITION_NAME	NUM_ROWS	AVG_ROW_LEN	LAST_ANALYZED
1	PAT_TREATMENT_MAP_PART	BLOCK8	0	0	05-MAY-21
2	PAT_TREATMENT_MAP_PART	BLOCK7	0	0	05-MAY-21
3	PAT_TREATMENT_MAP_PART	BLOCK6	0	0	05-MAY-21
4	PAT_TREATMENT_MAP_PART	BLOCK5	182	97	05-MAY-21
5	PAT_TREATMENT_MAP_PART	BLOCK4	895	100	05-MAY-21
6	PAT_TREATMENT_MAP_PART	BLOCK3	838	97	05-MAY-21
7	PAT_TREATMENT_MAP_PART	BLOCK2	806	95	05-MAY-21
8	PAT_TREATMENT_MAP_PART	BLOCK1	726	99	05-MAY-21

STEP 4 - Without partition

```
SELECT
        pr.patient_id,
        pr.pfname,
        plname,
        pr.patient_status,
        ad.appointment_id,
        ad.appt_date,
        ad.appt_time,
        ad.appt_status,
        ptm.med_name
FROM
        patient_records pr,
        appointment_dtls ad,
        patient_treatment_mapping ptm
WHERE pr.patient_id = ad.patient_id
AND ad.appointment_id = ptm.appointment_id
AND lower(ptm.med_name) LIKE '%en%'
AND ptm.treatment_id < 10;
```

On executing the above query, we see that the records are fetched in 0.193 seconds and has the below explain plan.

SQL | All Rows Fetched: 291 in 0.193 seconds

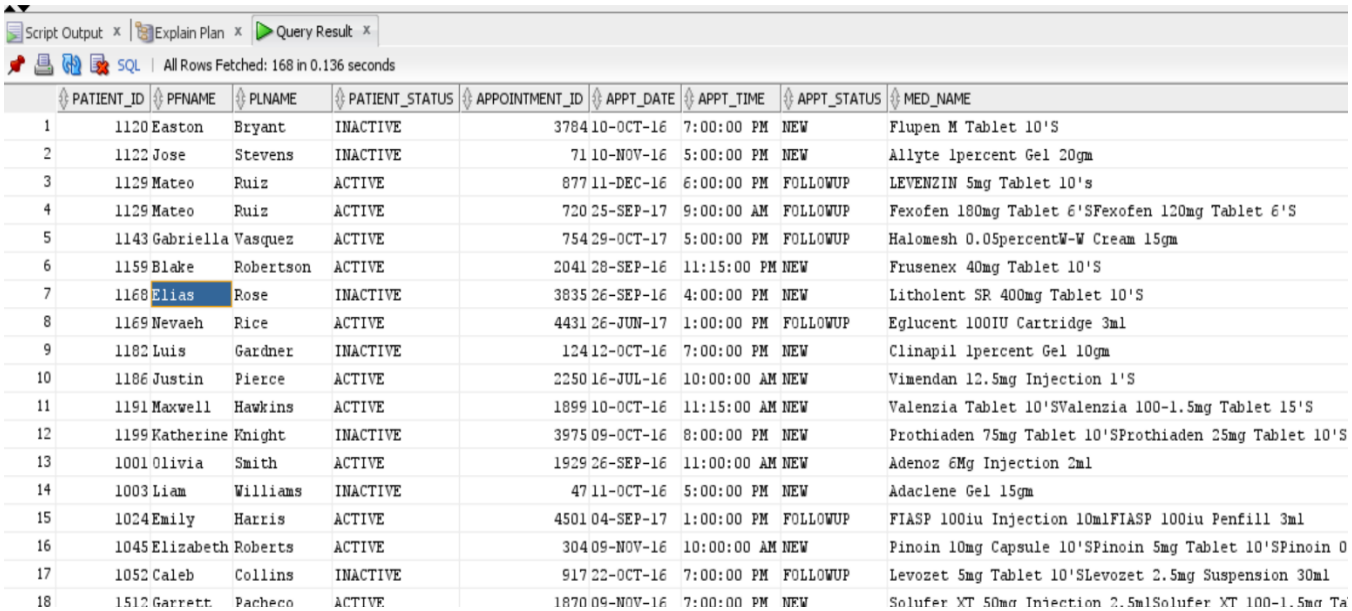
	PATIENT_ID	PFNAME	PLNAME	PATIENT_STATUS	APPOINTMENT_ID	APPT_DATE	APPT_TIME	APPT_STATUS	MED_NAME
1	1120	Easton	Bryant	INACTIVE	3784	10-OCT-16	7:00:00 PM	NEW	Flupen M Tablet 10'S
2	1122	Jose	Stevens	INACTIVE	71	10-NOV-16	5:00:00 PM	NEW	Allyte 1percent Gel 20gm
3	1129	Mateo	Ruiz	ACTIVE	877	11-DEC-16	6:00:00 PM	FOLLOWUP	LEVENZIN 5mg Tablet 10's
4	1129	Mateo	Ruiz	ACTIVE	720	25-SEP-17	9:00:00 AM	FOLLOWUP	Fexofen 180mg Tablet 6'SFexofen 120mg Tablet 6'S
5	1139	Sadie	Tucker	ACTIVE	2102	10-OCT-16	10:00:00 AM	NEW	Medapine AC Gel 15gmMedapine 1percent Gel
6	1143	Gabriella	Vasquez	ACTIVE	754	29-OCT-17	5:00:00 PM	FOLLOWUP	Medapine AC Gel 15gmMedapine 1percent Gel
7	1149	Madelyn	Mason	INACTIVE	1894	25-SEP-16	6:00:00 PM	NEW	Medapine AC Gel 15gmMedapine 1percent Gel
8	1157	Hailey	Hunt	INACTIVE	2651	25-OCT-17	1:00:00 PM	FOLLOWUP	Medapine AC Gel 15gmMedapine 1percent Gel
9	1159	Blake	Robertson	ACTIVE	2041	28-SEP-16	11:15:00 PM	NEW	Frusenex 40mg Tablet 10'S
10	1168	Elias	Rose	INACTIVE	3835	26-SEP-16	4:00:00 PM	NEW	Litholent SR 400mg Tablet 10'S
11	1169	Nevaeh	Rice	ACTIVE	4431	26-JUN-17	1:00:00 PM	FOLLOWUP	Eglucent 100IU Cartridge 3ml
12	1182	Luis	Gardner	INACTIVE	568	26-APR-17	6:00:00 PM	FOLLOWUP	Medapine AC Gel 15gmMedapine 1percent Gel
13	1182	Luis	Gardner	INACTIVE	124	12-OCT-16	7:00:00 PM	NEW	Clinapil 1percent Gel 10gm
14	1186	Justin	Pierce	ACTIVE	2250	16-JUL-16	10:00:00 AM	NEW	Medapine AC Gel 15gmMedapine 1percent Gel

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			77	120
HASH JOIN			77	120
Access Predicates				
PR.PATIENT_ID=AD.PATIENT_ID				
NESTED LOOPS			77	120
NESTED LOOPS				
STATISTICS COLLECTOR				
HASH JOIN			77	90
Access Predicates				
AD.APPOINTMENT_ID=PTM.APPOINTMENT_ID				
NESTED LOOPS			77	90
STATISTICS COLLECTOR				
TABLE ACCESS	PATIENT_TREATMENT_MAPPING	FULL	77	22
Filter Predicates				
AND				
PTM.TREATMENT_ID<10				
LOWER(PTM.MED_NAME) LIKE '%en%'				
TABLE ACCESS	APPOINTMENT_DTLS	BY INDEX ROWID	1	68
INDEX	APPOINTMENT_DTLS_PK	UNIQUE SCAN		
Access Predicates				
AD.APPOINTMENT_ID=PTM.APPOINTMENT_ID				
TABLE ACCESS	APPOINTMENT_DTLS	FULL	10000	68
INDEX	PATIENT_RECORDS_PK	UNIQUE SCAN		
Access Predicates				
PR.PATIENT_ID=AD.PATIENT_ID				
TABLE ACCESS	PATIENT_RECORDS	BY INDEX ROWID	1	30
TABLE ACCESS	PATIENT_RECORDS	FULL	4135	30

STEP 5 - With partition

```
SELECT
        pr.patient_id,
        pr.pfname,
        plname,
        pr.patient_status,
        ad.appointment_id,
        ad.appt_date,
        ad.appt_time,
        ad.appt_status,
        ptm.med_name
FROM
        patient_records pr,
        appointment_dtls ad,
        pat_treatment_map_part ptm
WHERE pr.patient_id = ad.patient_id
AND ad.appointment_id = ptm.appointment_id
AND lower(ptm.med_name) LIKE '%en%'
AND ptm.treatment_id < 10;
```

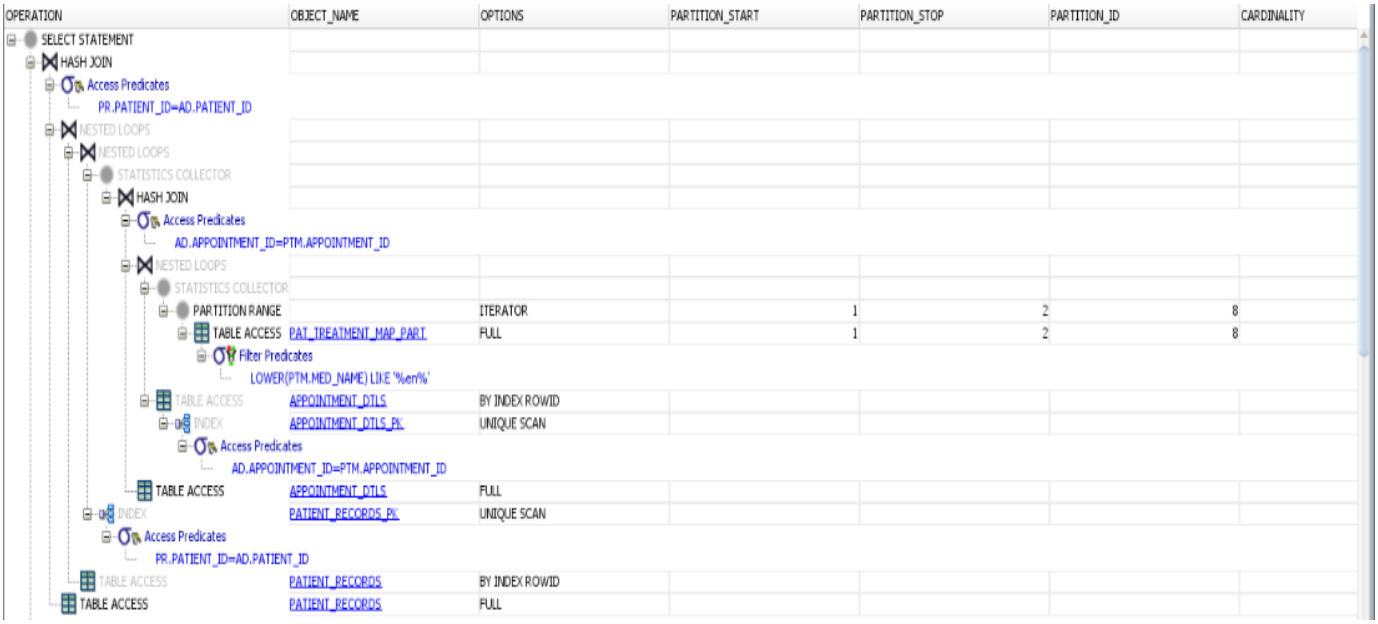
On executing the above query, we see that the records are fetched in 0.136 seconds and has the below explain plan that shows the use of partition.


Script Output | Explain Plan | Query Result

SQL | All Rows Fetched: 168 in 0.136 seconds

	PATIENT_ID	PFNAME	PLNAME	PATIENT_STATUS	APPOINTMENT_ID	APPT_DATE	APPT_TIME	APPT_STATUS	MED_NAME
1	1120	Easton	Bryant	INACTIVE	3784	10-OCT-16	7:00:00 PM	NEW	Flupen M Tablet 10'S
2	1122	Jose	Stevens	INACTIVE	71	10-NOV-16	5:00:00 PM	NEW	Allyte 1percent Gel 20gm
3	1129	Mateo	Ruiz	ACTIVE	877	11-DEC-16	6:00:00 PM	FOLLOWUP	LEVENZIN 5mg Tablet 10's
4	1129	Mateo	Ruiz	ACTIVE	720	25-SEP-17	9:00:00 AM	FOLLOWUP	Fexofen 180mg Tablet 6'SFexofen 120mg Tablet 6'S
5	1143	Gabriella	Vasquez	ACTIVE	754	29-OCT-17	5:00:00 PM	FOLLOWUP	Halomesh 0.05percentW-W Cream 15gm
6	1159	Blake	Robertson	ACTIVE	2041	28-SEP-16	11:15:00 PM	NEW	Frusenex 40mg Tablet 10'S
7	1168	Elias	Rose	INACTIVE	3835	26-SEP-16	4:00:00 PM	NEW	Litholent SR 400mg Tablet 10'S
8	1169	Nevaeh	Rice	ACTIVE	4431	26-JUN-17	1:00:00 PM	FOLLOWUP	Eglucent 100IU Cartridge 3ml
9	1182	Luis	Gardner	INACTIVE	124	12-OCT-16	7:00:00 PM	NEW	Clinapil 1percent Gel 10gm
10	1186	Justin	Pierce	ACTIVE	2250	16-JUL-16	10:00:00 AM	NEW	Vimendan 12.5mg Injection 1'S
11	1191	Maxwell	Hawkins	ACTIVE	1899	10-OCT-16	11:15:00 AM	NEW	Valenzia Tablet 10'SValenzia 100-1.5mg Tablet 15'S
12	1199	Katherine	Knight	INACTIVE	3975	09-OCT-16	8:00:00 PM	NEW	Prothiaden 75mg Tablet 10'SProthiaden 25mg Tablet 10'S
13	1001	Olivia	Smith	ACTIVE	1929	26-SEP-16	11:00:00 AM	NEW	Adenoz 6Mg Injection 2ml
14	1003	Liam	Williams	INACTIVE	47	11-OCT-16	5:00:00 PM	NEW	Adaclene Gel 15gm
15	1024	Emily	Harris	ACTIVE	4501	04-SEP-17	1:00:00 PM	FOLLOWUP	FIASP 100iu Injection 10mlFIASP 100iu Penfill 3ml
16	1045	Elizabeth	Roberts	ACTIVE	304	09-NOV-16	10:00:00 AM	NEW	Pinoin 10mg Capsule 10'SPinoin 5mg Tablet 10'SPinoin 0
17	1052	Caleb	Collins	INACTIVE	917	22-OCT-16	7:00:00 PM	FOLLOWUP	Levozet 5mg Tablet 10'SLevozet 2.5mg Suspension 30ml
18	1512	Garrett	Pacheco	ACTIVE	1870	09-NOV-16	7:00:00 PM	NEW	Solufer XT 50mg Injection 2.5mlSolufer XT 100-1.5mg Ta

Explain Plan for the above query



OPERATION	OBJECT_NAME	OPTIONS	PARTITION_START	PARTITION_STOP	PARTITION_ID	CARDINALITY
SELECT STATEMENT						
HASH JOIN						
Access Predicates						
PR.PATIENT_ID=AD.PATIENT_ID						
NESTED LOOPS						
NESTED LOOPS						
STATISTICS COLLECTOR						
HASH JOIN						
Access Predicates						
AD.APPOINTMENT_ID=PTM.APPOINTMENT_ID						
NESTED LOOPS						
STATISTICS COLLECTOR						
PARTITION RANGE		ITERATOR	1	2		8
TABLE ACCESS	PAT_TREATMENT_MAP_PART	FULL	1	2		8
Filter Predicates						
LOWER(PTM.MED_NAME) LIKE '%en%'						
TABLE ACCESS	APPOINTMENT_DTLS	BY INDEX ROWID				
INDEX	APPOINTMENT_DTLS_PK	UNIQUE SCAN				
Access Predicates						
AD.APPOINTMENT_ID=PTM.APPOINTMENT_ID						
TABLE ACCESS	APPOINTMENT_DTLS	FULL				
INDEX	PATIENT_RECORDS_PK	UNIQUE SCAN				
Access Predicates						
PR.PATIENT_ID=AD.PATIENT_ID						
TABLE ACCESS	PATIENT_RECORDS	BY INDEX ROWID				
TABLE ACCESS	PATIENT_RECORDS	FULL				

OTHERS

DBA SCRIPTS

This section covers the scripts that are useful for Database Administrators in managing and monitoring database operations.

QUERY 1 - Display SQL statements for the current database sessions.

```
SELECT
   vs.sid,
   vs.status,
   vs.process,
   vs.schemaname,
   vs.osuser,
   va.sql_text
FROM
   v$session vs,
   v$sqlarea va
WHERE vs.sql_hash_value = va.hash_value
AND vs.sql_address = va.address;
```

OUTPUT :

	SID	STATUS	PROCESS	SCHEMANAME	OSUSER	SQL_TEXT
1	38	INACTIVE	38884	EMERALD	chowd	select vendor_id, vendor_type,vendor_name from vendor_details order by vendor_name
2	45	INACTIVE	16244	EMERALD	SAMRUDHI_MAHATRE	select * from orders
3	74	INACTIVE	12740	DB289	gaura	SELECT /*+ no_parallel("BILLING") */ ROWID "ROWID", ORA_ROWSCN "ORA_ROWSCN", BILL_ID BILL_ID,
4	80	ACTIVE	16816	EMERALD	Purva	SELECT vs.sid, vs.status, vs.process, vs.schemaname, vs.osuser,
5	95	INACTIVE	9034	DBEPNDT	dberndt	SELECT /*+ no_parallel("PLOTS") */ ROWID "ROWID", ORA_ROWSCN "ORA_ROWSCN", MOVIE_GUID MOVIE_(

QUERY 2 - Display information on all active and inactive database sessions.

```
SELECT
   NVL(vs.username, '(oracle)') AS username,
   vs.osuser,
   vs.sid,
   vs.serial#,
   vs.lockwait,
   vs.status,
   vs.machine,
   vs.program,
   TO_CHAR(vs.logon_Time,'DD-MON-YYYY HH24:MI:SS') AS logon_time
FROM
    v$session vs
WHERE  vs.status in ('ACTIVE',’INACTIVE’)
ORDER BY vs.username, vs.osuser;
```

OUTPUT :

	USERNAME	OSUSER	SID	SERIAL#	LOCKWAIT	STATUS	MACHINE	PROGRAM	LOGON_TIME
1	DB289	gaura	74	29502	(null)	INACTIVE	DESKTOP-H1I0FI0	SQL Developer	06-MAY-2021 14:07:34
2	DB289	siddharthmundada	84	48483	(null)	INACTIVE	Siddharths-MacBook-Pro.local	SQL Developer	06-MAY-2021 12:36:01
3	DB341	David	54	17634	(null)	INACTIVE	DESKTOP-PIDT43J	SQL Developer	06-MAY-2021 10:33:53
4	DBERNDT	dberndt	95	7806	(null)	INACTIVE	ITS-MACs-Mini.home	SQL Developer	06-MAY-2021 11:54:37
5	EMERALD	Purva	80	52368	(null)	ACTIVE	DESKTOP-4NKDJK0	SQL Developer	06-MAY-2021 14:09:36
6	EMERALD	SAMRUDHI_MAHATRE	45	59229	(null)	INACTIVE	LAPTOP-G7LDP1AU	SQL Developer	06-MAY-2021 14:14:32
7	EMERALD	chowd	38	31631	(null)	INACTIVE	DESKTOP-DPJBSNT	SQL Developer	06-MAY-2021 14:10:53
8	EMERALD	user	94	33252	(null)	INACTIVE	DESKTOP-EM01MHJ	SQL Developer	06-MAY-2021 14:44:46
9	(oracle)	oracle	10	49777	(null)	ACTIVE	READE	ORACLE.EXE (VKRM)	26-JAN-2021 14:57:59
10	(oracle)	oracle	11	2918	(null)	ACTIVE	READE	ORACLE.EXE (DIA0)	26-JAN-2021 14:57:59

QUERY 3 - Display a list of top 10 SQL statements that are using the most resources.

```
SELECT
   SUBSTR(va.sql_text,1,50) sql_text,
   TRUNC(va.disk_reads/DECODE(va.executions,0,1,va.executions)) reads_per_execution,
   va.buffer_gets,
   va.disk_reads,
   va.executions,
   va.sorts,
   va.address
FROM
   v$sqlarea va
WHERE ROWNUM <=10
ORDER BY 2 DESC;
```

OUTPUT :

	SQL_TEXT	READS_PER_EXECUTION	BUFFER_GETS	DISK_READS	EXECUTIONS	SORTS	ADDRESS
1	select extract(day from retention)*24*60*60 +	1	10	1	1	0	000007FF43163408
2	SELECT /* DS_SVC */ /*+ dynamic_sampling(0) no_sql	0	1783	0	1	0	000007FF30478B98
3	SELECT NVL(TO_NUMBER(EXTRACT(XMLTYPE(:B2), :B1))	0	27	0	42	0	000007FF43C25D20
4	select exptime, ltime, astatus, lcount from user$	0	313	7	75	0	000007FF5EF686B0
5	SELECT /* DS_SVC */ /*+ dynamic_sampling(0) no_sql	0	199	0	1	0	000007FF2B977998
6	insert into wrh$_osstat (dbid, con_dbid, snap_id	0	373	16	23	0	000007FF5D479C58
7	SELECT /*+ CONNECT_BY_FILTERING */ s.privilege# FR	0	306	4	39	39	000007FF5D1CB8D8
8	select * from dba_tab_partitions where table_owner	0	9441	0	1	3	000007FF30DD2648
9	select * from rmdb.movies where movie_title like '	0	90	0	2	0	000007FF57C707B8
10	SELECT /* DS_SVC */ /*+ dynamic_sampling(0) no_sql	0	140	0	7	0	000007FF59A3E960

QUERY 4 - Display a list of all roles and privileges granted.

```
SELECT
   rr.role,
   rr.granted_role,
   rr.admin_option
FROM
   role_role_privs rr
ORDER BY
   rr.role,
   rr.granted_role;
```

OUTPUT :

	ROLE	GRANTED_ROLE	ADMIN_OPTION
1	SELECT_CATALOG_ROLE	HS_ADMIN_SELECT_ROLE	NO
2	USF_SELECT	SELECT_CATALOG_ROLE	NO
3	USF_STUDENT	MMRKT_TRADER	NO
4	USF_STUDENT	USF_CONNECT	NO
5	USF_STUDENT	USF_CREATE	NO
6	USF_STUDENT	USF_DEBUG	NO
7	USF_STUDENT	USF_EXECUTE	NO
8	USF_STUDENT	USF_SELECT	NO
9	USF_STUDENT	USF_TUNE	NO

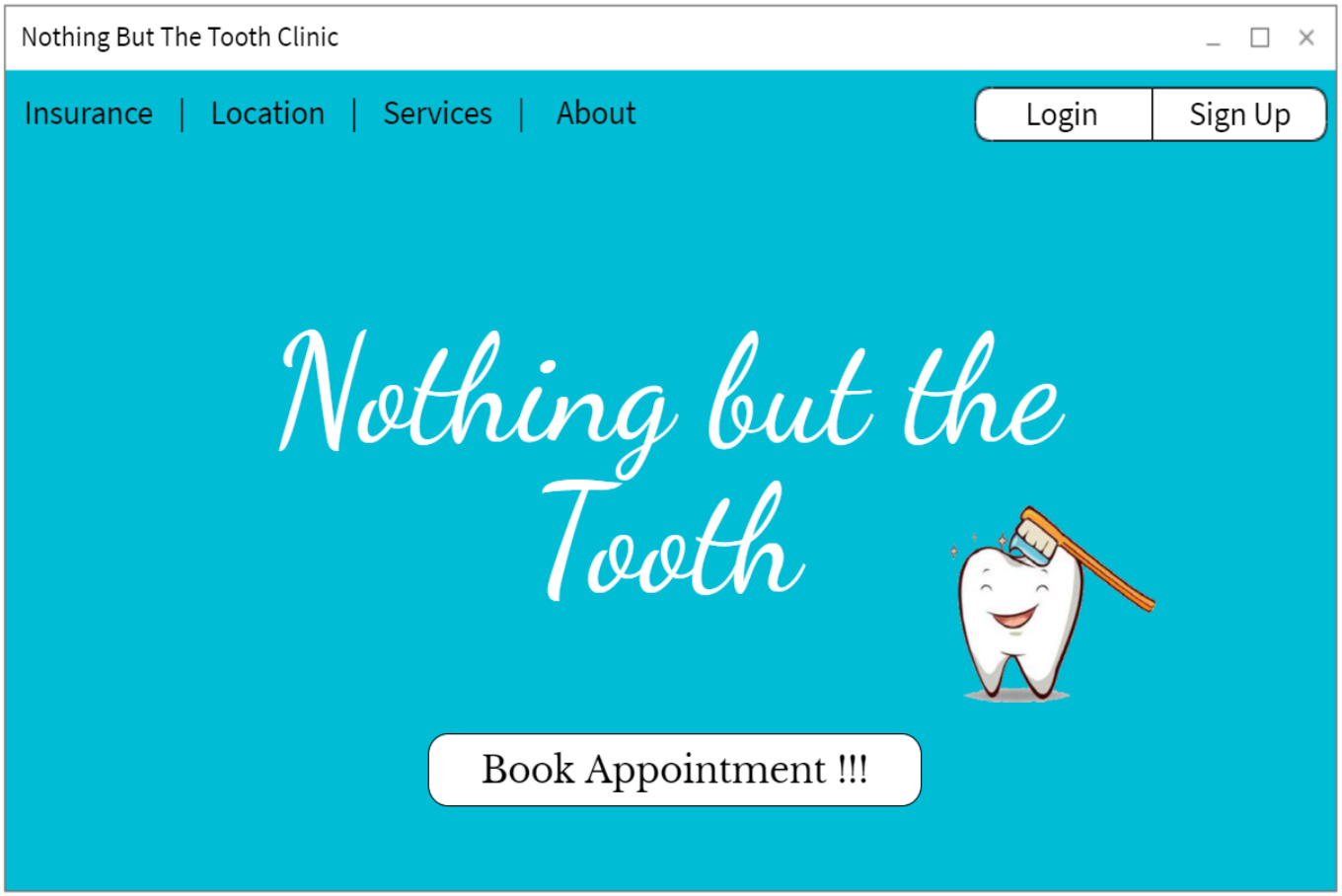
```
SELECT
   rp.grantee,
   rp.granted_role,
   rp.admin_option,
   rp.default_role
FROM
   dba_role_privs rp
ORDER BY
   rp.grantee,
   rp.granted_role;
```

	GRANTEE	GRANTED_ROLE	ADMIN_OPTION	DEFAULT_ROLE
1	ABERNDT	USF_STUDENT	NO	YES
2	AP	RESOURCE	NO	NO
3	AP	USF_STUDENT	NO	YES
4	APEX_040200	CONNECT	NO	YES
5	APEX_040200	RESOURCE	NO	YES
6	AQUA	USF_STUDENT	NO	YES
7	BEERDB	RESOURCE	NO	NO
8	BEERDB	USF_STUDENT	NO	YES
9	BI	RESOURCE	NO	YES
10	BLUE	USF_STUDENT	NO	YES
11	CTXSYS	CTXAPP	YES	YES
12	CTXSYS	RESOURCE	NO	YES
13	DATAPUMP_EXP_FULL_DATABASE	EXP_FULL_DATABASE	NO	YES

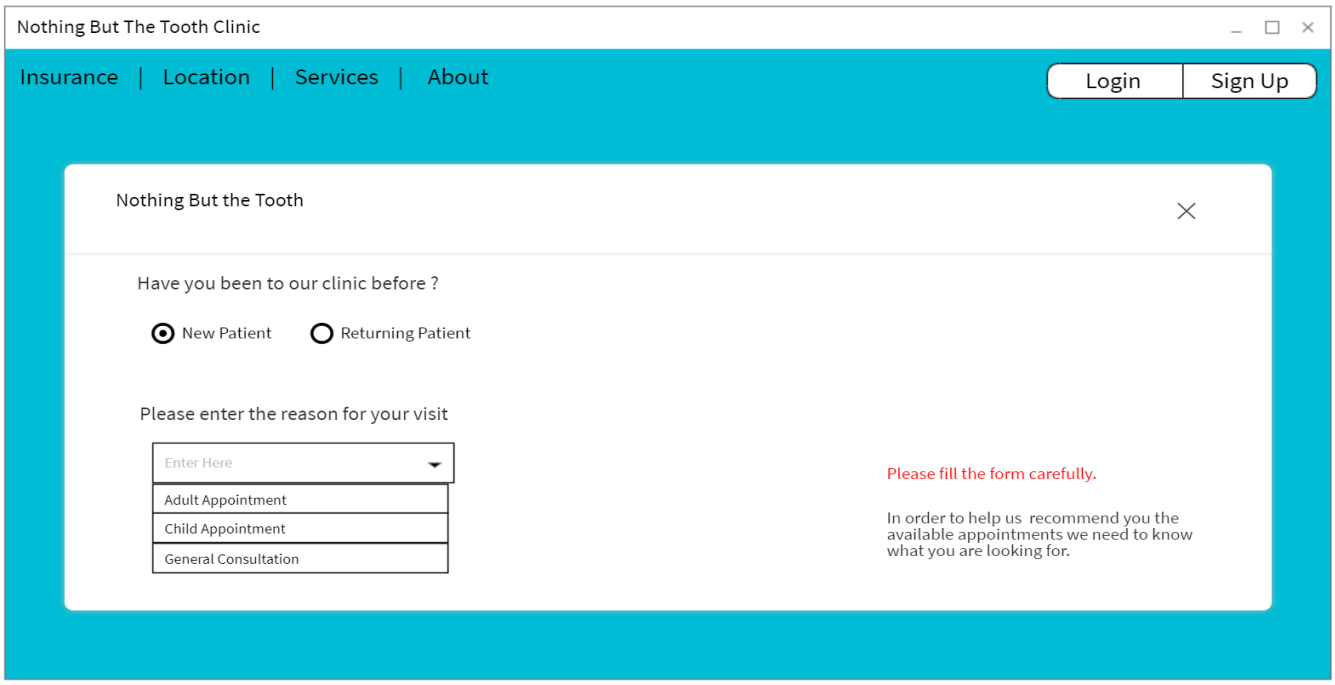
USER INTERFACE MOCKUPS

For the demo purpose we are showing Interface mockups for Appointment module and its related activities.

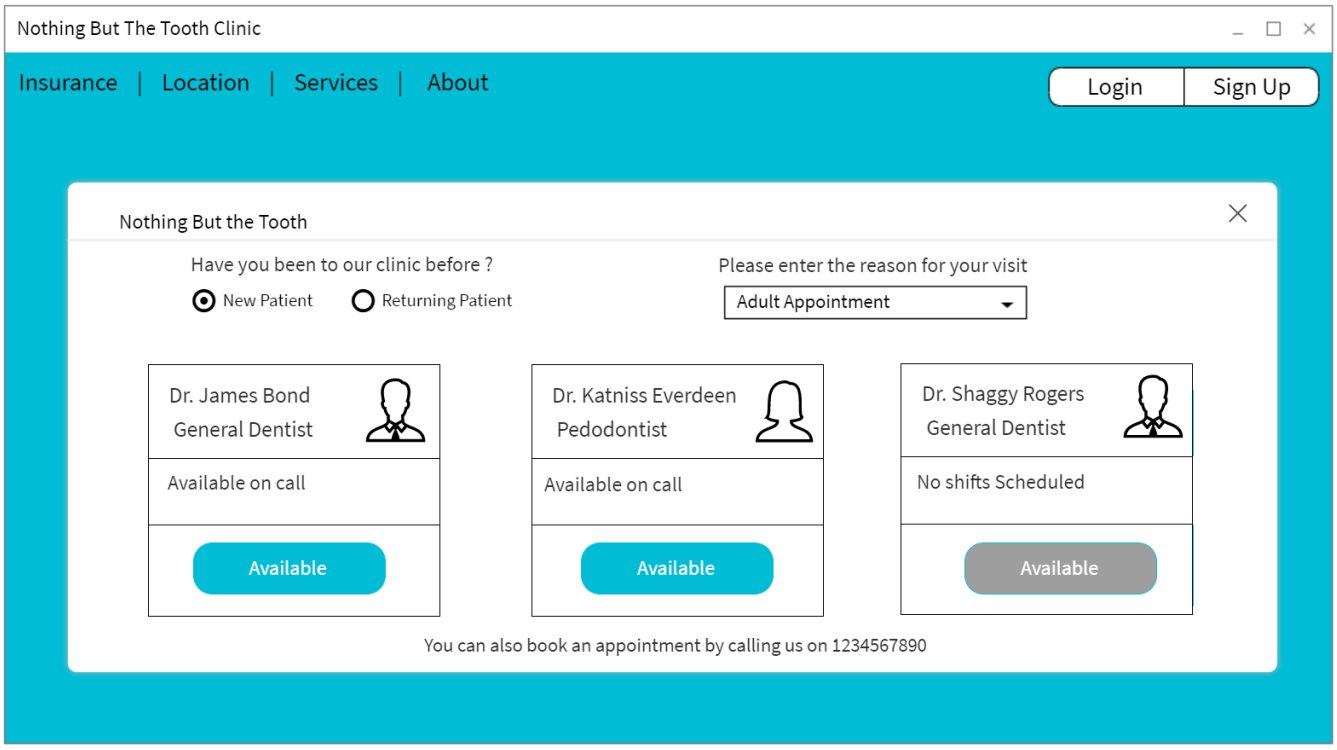
HOME PAGE - This page is visible when the user tries to open the clinic's website. Here the user can either Login, Sign Up or directly look for the available appointments and book one.



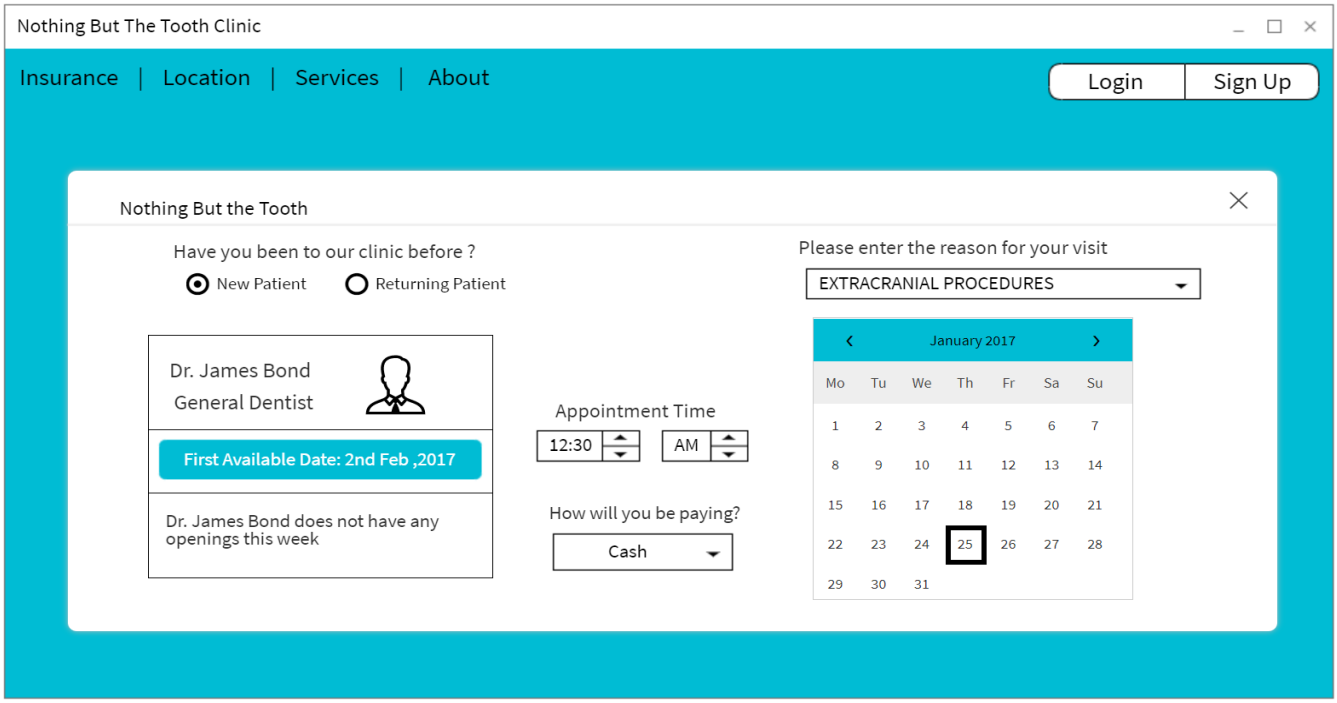
BOOK APPOINTMENT -After clicking on the Book Appointment button on the Home page, the user is navigated to this page.



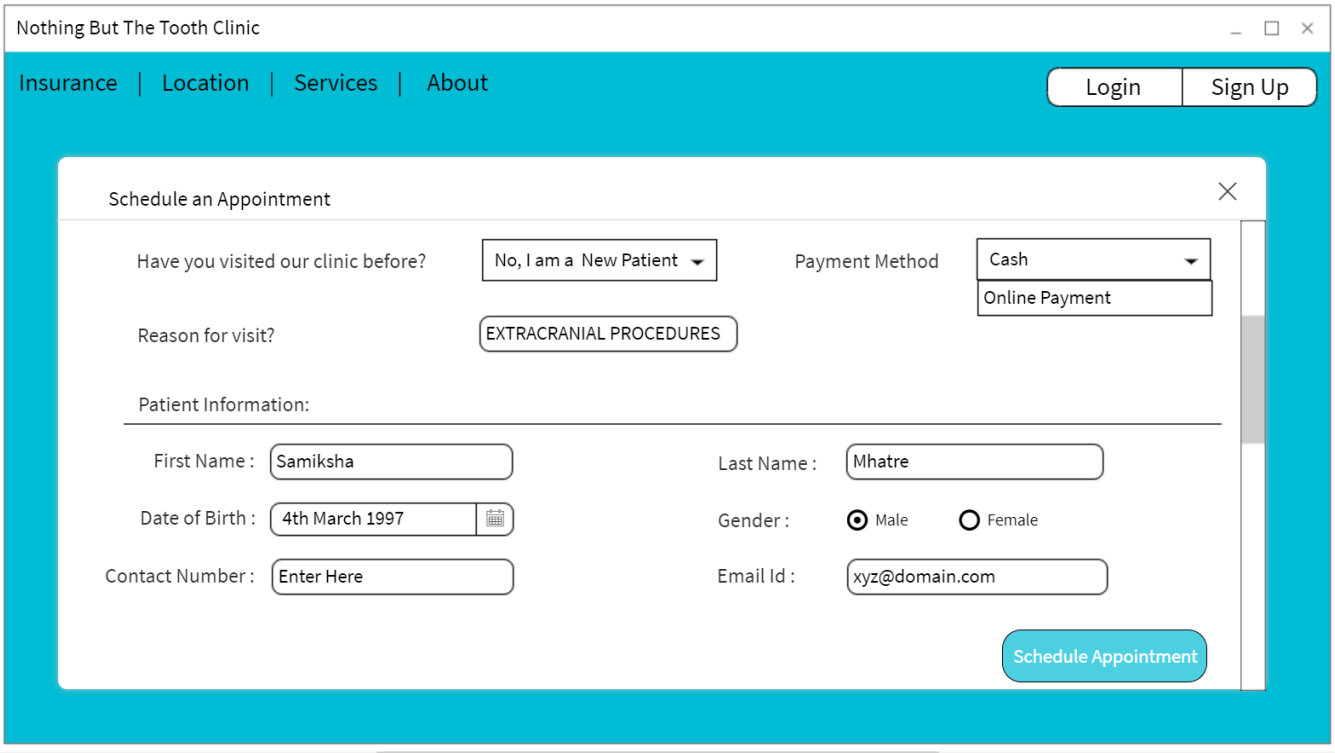
Once the user fills in the required information, they are directed to this page where they can view a list of available dentists.



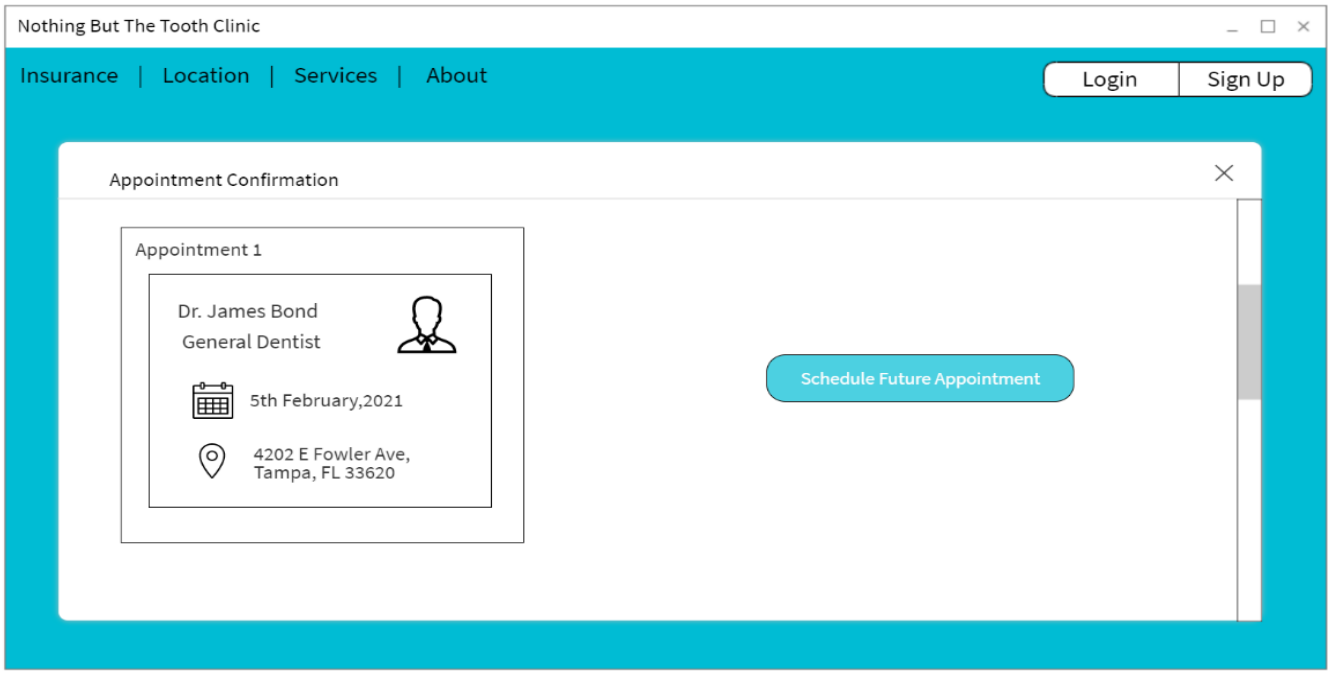
Once the user selects the dentist of their choice, they are directed to this page where they can check their available appointment date and time for the appointment.



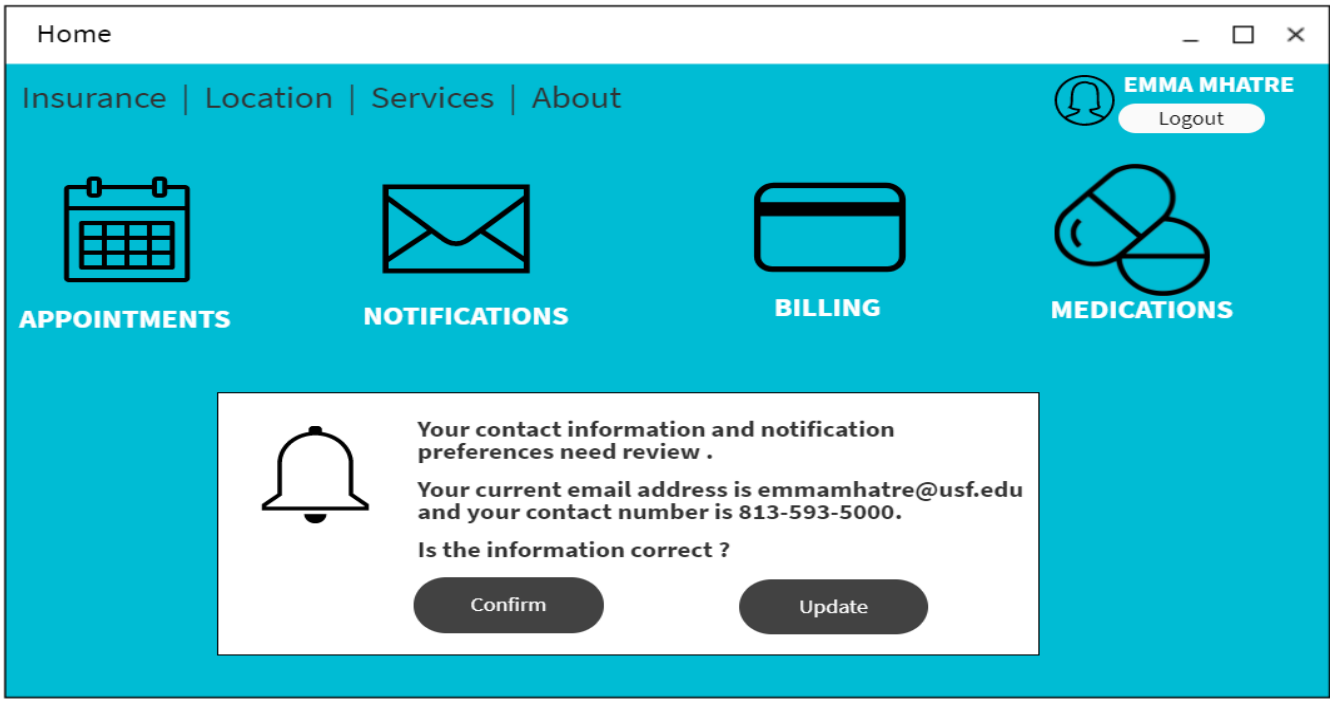
After selecting the date, time and the payment type, the patient is directed to a page where he is asked to fill his personal details.



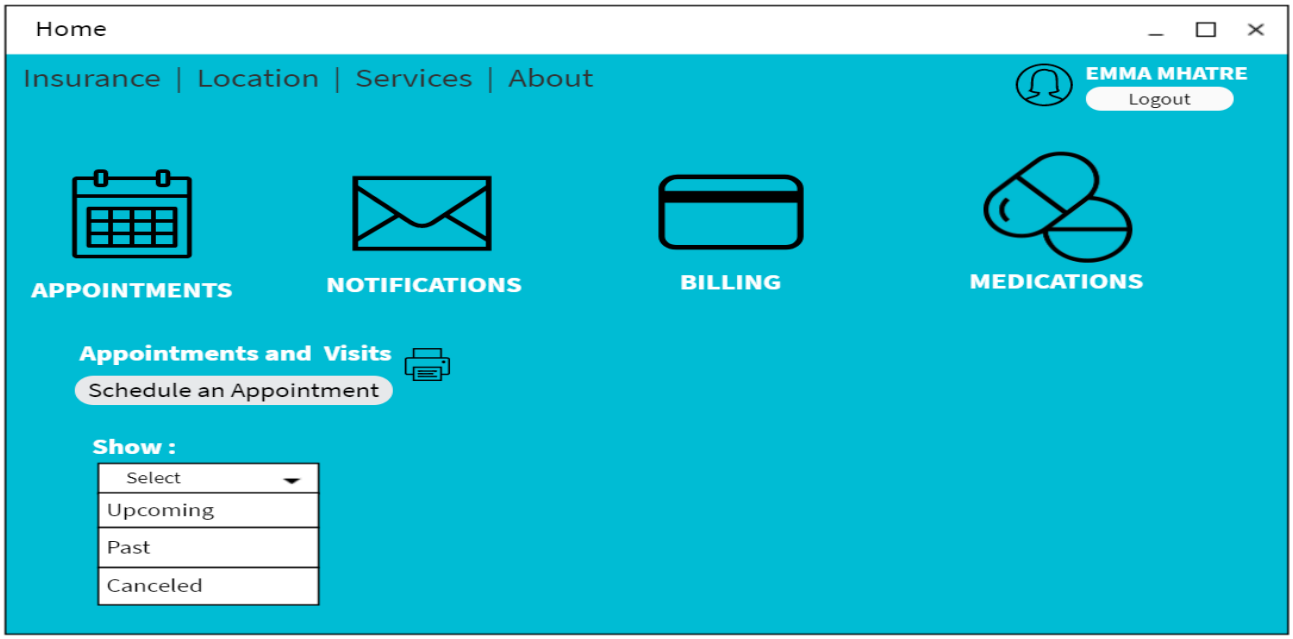
APPOINTMENT CONFIRMATION -Once the user has entered all their personal details, they will be navigated to the appointment confirmation page, where they get to look at recently scheduled appointment and book a future appointment in advance if required.



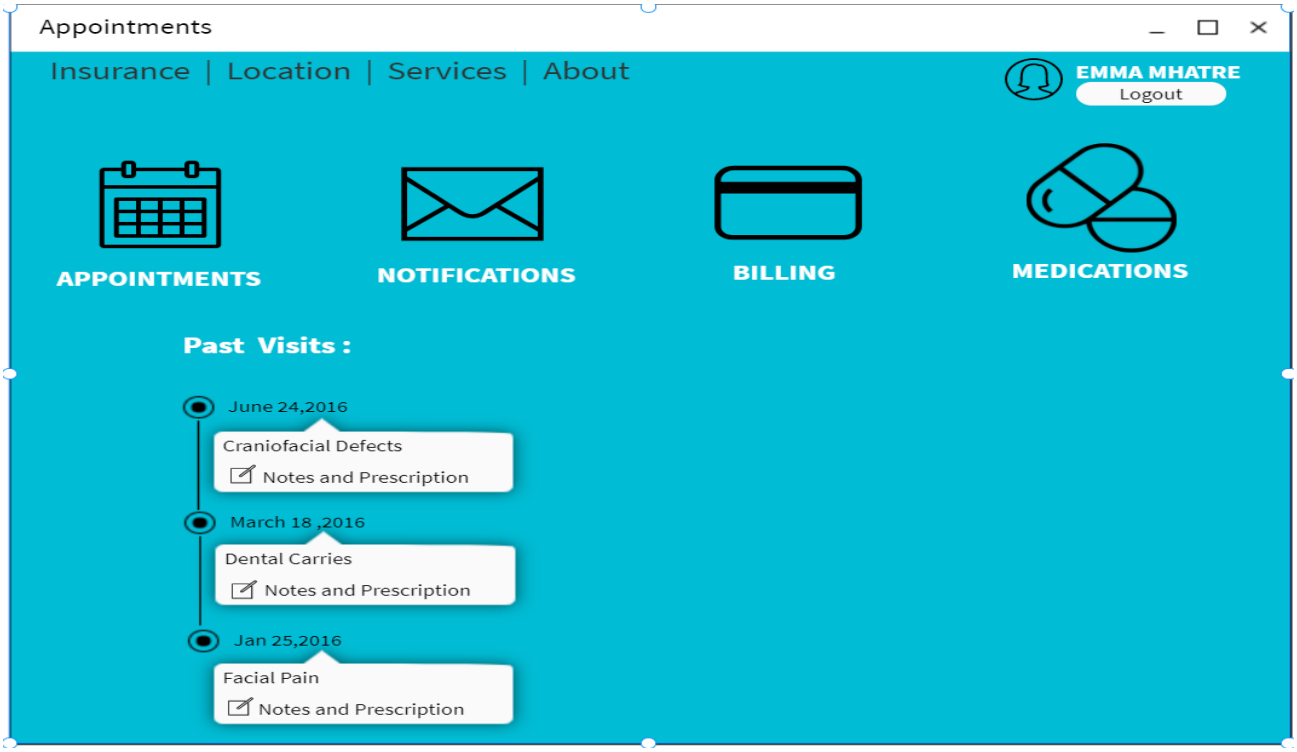
HOME PAGE AFTER LOGIN - Once the user has logged into the system successfully, a pop-up notification will be displayed to confirm or update the details entered.



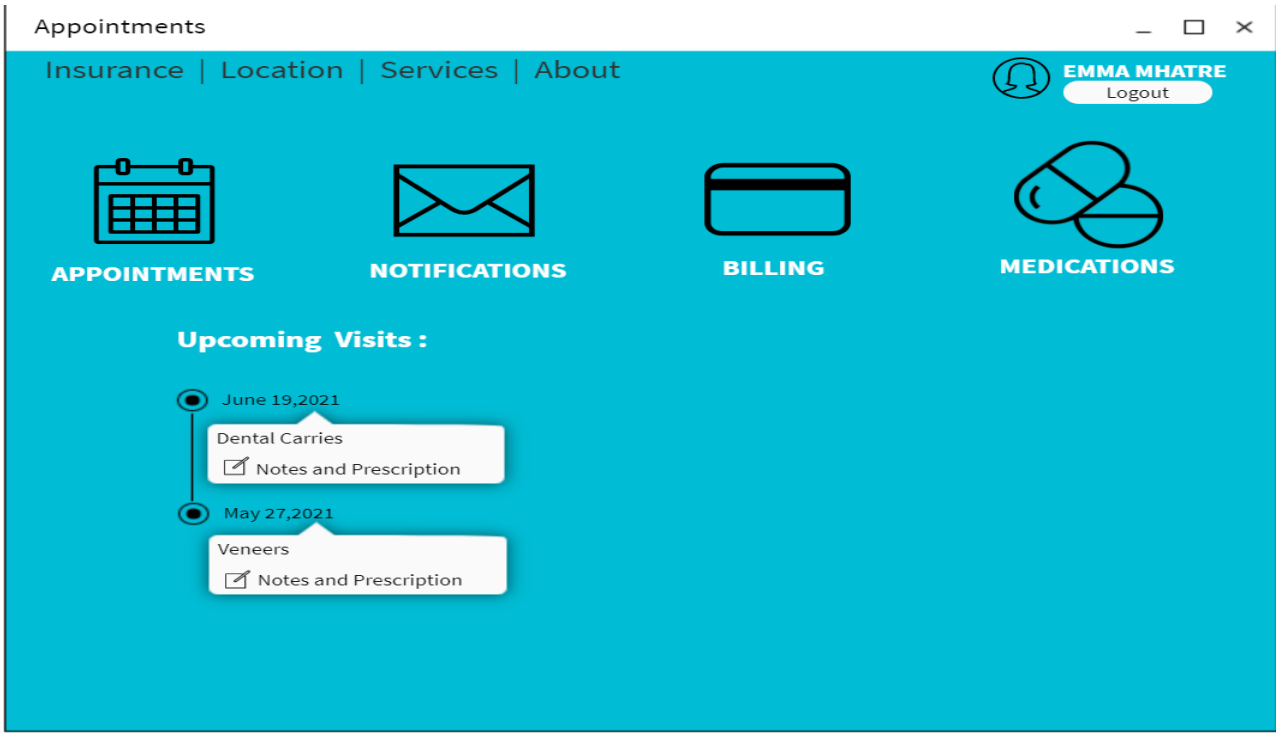
APPOINTMENT SCHEDULING - User can schedule a new appointment according to their convenience and can have a track of past and upcoming clinic visits.



VIEW PAST APPOINTMENTS – User can keep a track of past visits and can access previous notes and medical prescriptions.



VIEW UPCOMING APPOINTMENTS - User can view a list of future appointments.



DATA VISUALIZATION

In order to analyze, draw conclusions and develop recommendations we have done few visualizations using PowerBI. The input data was provided to the tool is from the sample data we created for the project.

DASHBOARD

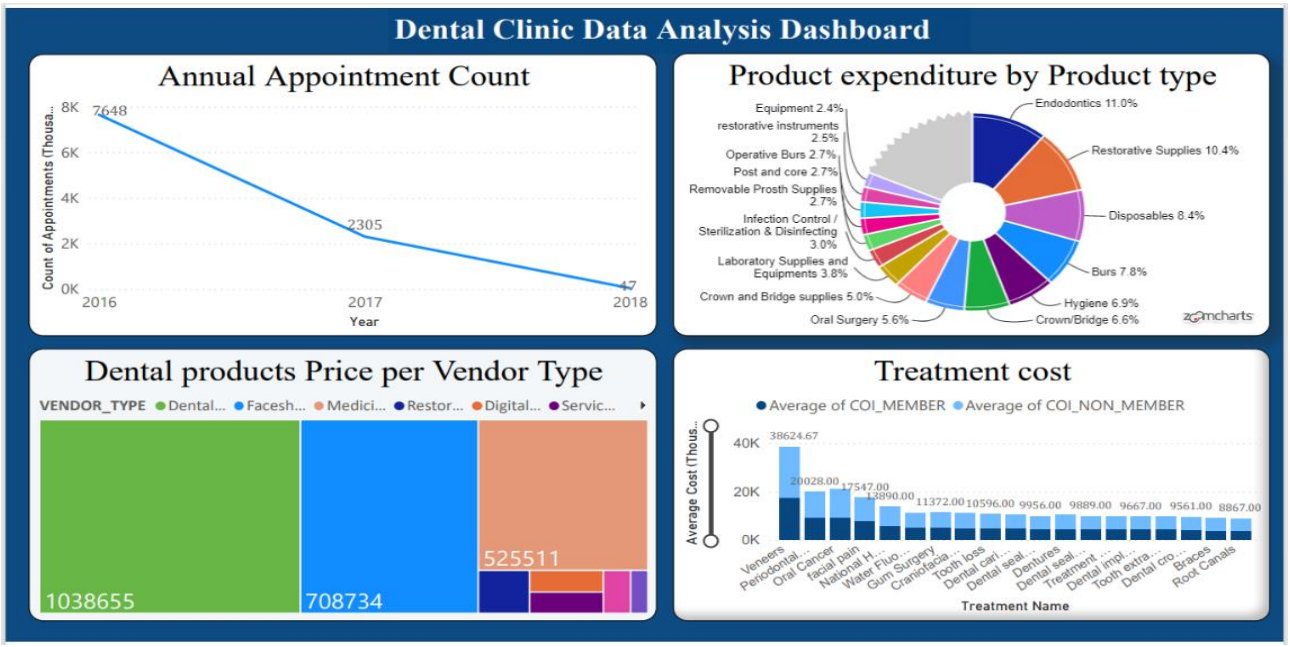


CHART 1: Count of Appointments on yearly basis

This chart displays the count of total appointments taken by the patients in the respective years. This is implemented using Line chart.

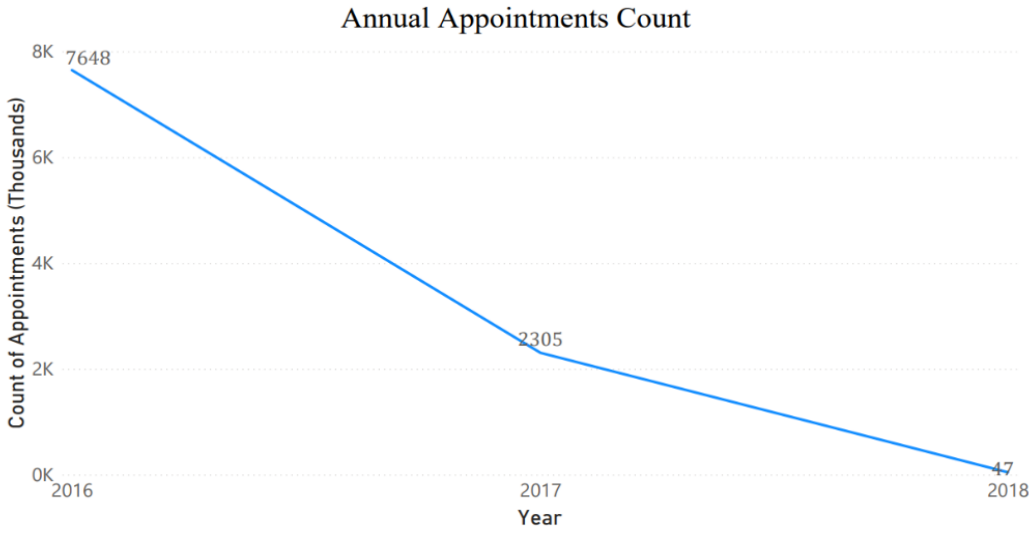
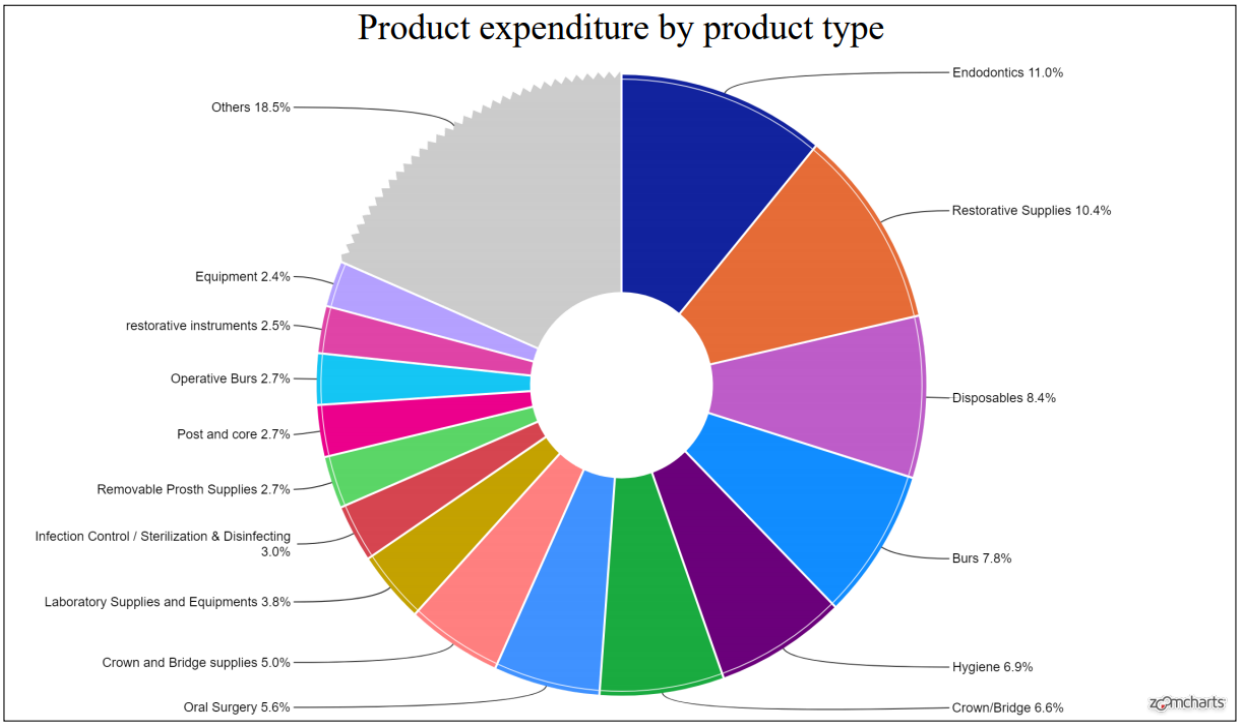


CHART 2: Product expenditure by the type of dental product.

This chart depicts the expenditure of dental products in the drill down manner using Donut chart. Firstly, it displays the product expenditure by product type as shown below. This chart lets one know how much amount is invested in the dental products of the clinic.



On clicking on either of the product types, it leads to its child field-Product Name as shown below- Along with the cost of products, it also gives the percentage of total cost each product costs.

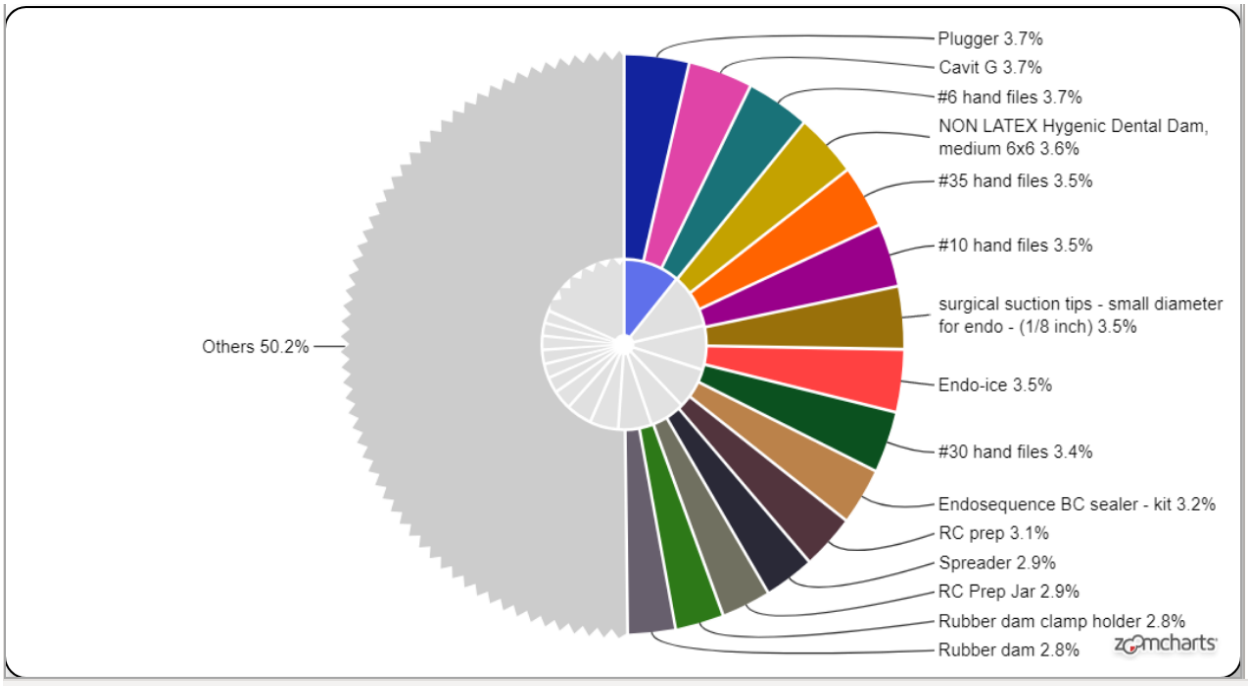


CHART 3: Dental Products as per Vendor Type

This chart displays the total product price invested for each Vendor. This shows which vendor is looked upon at most looking at the budget invested in buying products from that vendor.

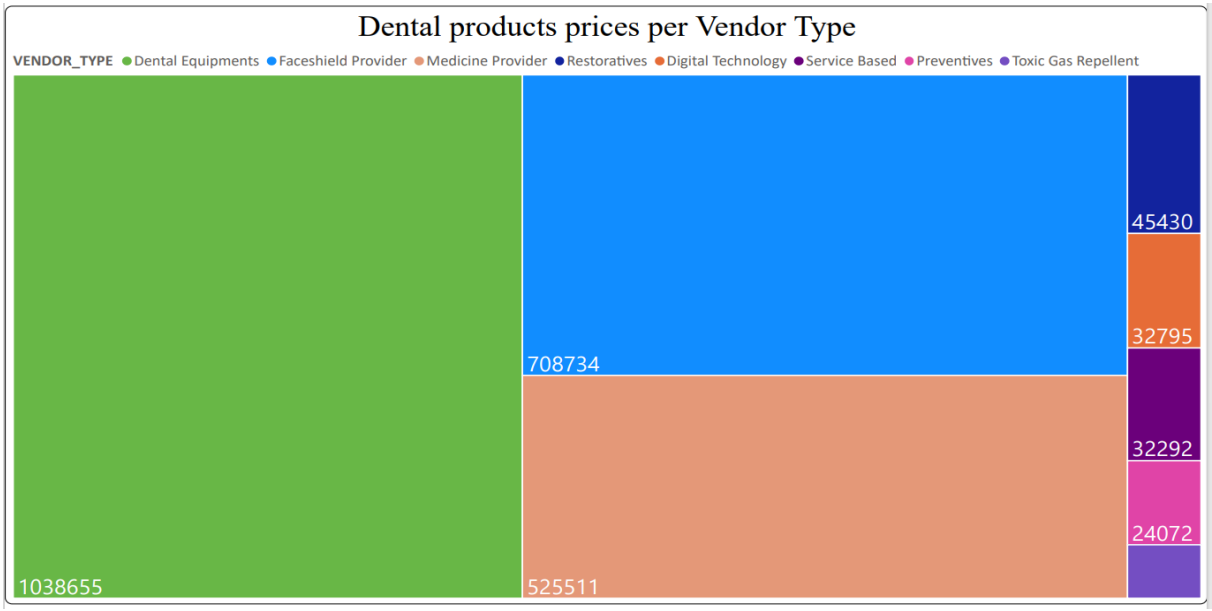
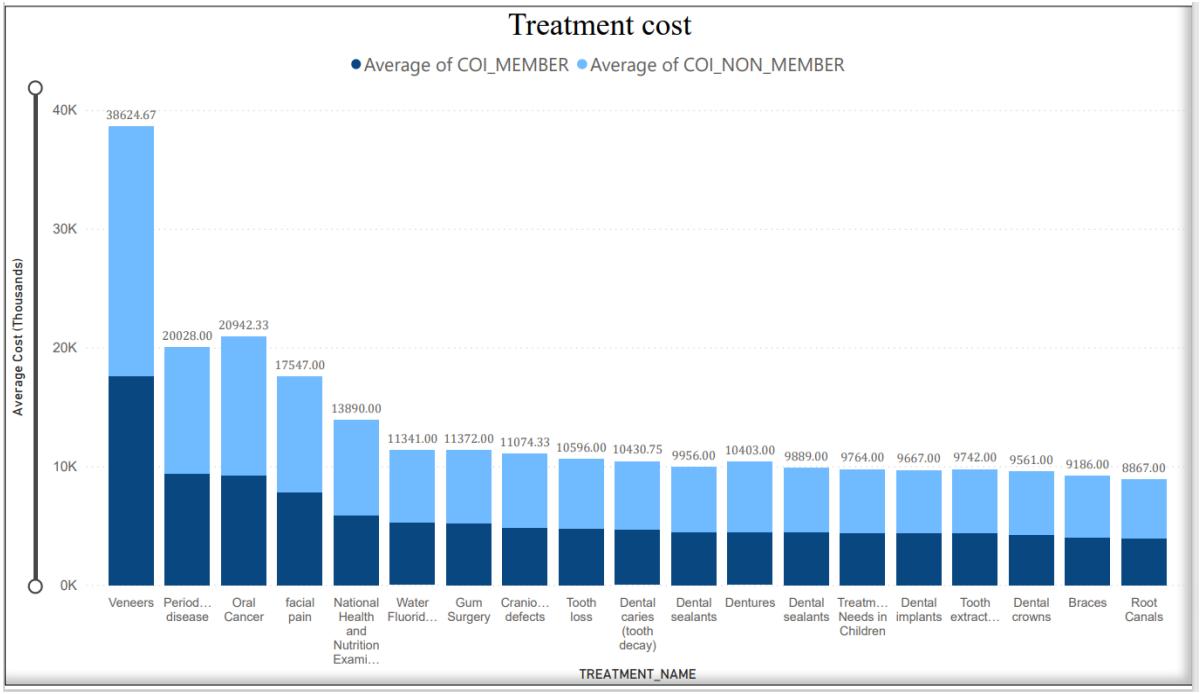


CHART 4: Treatment Cost

This chart displays the average cost of treatment of Insured members vs non-insured members.



For viewing the complete dashboard and reports, please visit the below link-

https://app.powerbi.com/groups/me/dashboards/985350ba-31a9-4756-a198-5eb8f5860781?ctid=741bf7de-e2e5-46df-8d67-82607df9deaa

DB SECURITY

In the dental clinic database system, the access privileges and security management are done by the system administrators. To access the website and the backend database, users must be authenticated. One can access the overall system by their respective credentials (username and password).

There are different access roles given to different types of user, as shown below-

Roles	Access and Privileges
Database Administrator	This user will have the admin access and control to the entire system. This user can create, edit, update, delete records on the website.
User Role	This is the default access given to every user in clinic using this system. They will have access to various functionalities to operate on. It is further divided into multiple types of users.
Patient Role	This will be assigned to all the patients registered in the system. They have read/write access on their personal and appointment details. They can see medicine prescription and notes as well.
Vendor	This will be assigned to all the vendors registered in the system. They will have all the read/write access to their product details and personal information.
Dentist	They have read/write access to all the patient, appointments and treatment details.
Assistant	They have read access to all the patient, appointments and treatment details.
Lab Technician	They have read access to all the patient details and update lab reports when required.
Staff	They have write access to all the inventory operations.
Receptionist	They have write access to patient details and their appointment details and are responsible for billing and printing reports.

ARCHIVAL PROCESS

Systems may have data that is no longer used actively but is important and can be used to retain for future reference. This type of data being on primary storage tables may increase the retrieval time of records when queried on it for current scenarios. In order to optimize the retrieval, we archive such records to secondary storage i.e tables that are access when required and remove them from the primary tables.

To demonstrate the process, we have archived **LOGIN_DETAILS** table here for few records. Following are the steps followed to achieve the Archival.

STEP 1: The below screenshot shows the records on which archival would be demonstrated.

```
SELECT * FROM LOGIN_DETAILS
WHERE TRUNC(LOGIN_DATE_TIME) < SYSDATE
AND LOGIN_ID IN (50101,50102,50103,50104,50105,50106,50107,50110);
```

Script Output | Query Result | SQL | All Rows Fetched: 8 in 0.025 seconds

	LOGIN_ID	USER_NAME	PASSWORD	LOGIN_DATE_TIME
1	50101	Lyam_COLEMAN	Lyam_COLEMAN803	17-APR-21 05.50.36.000000000 PM
2	50102	Phineas_SIMMONS	Phineas_SIMMONS791	04-APR-18 01.34.44.000000000 AM
3	50103	Riker_PATTERSON	Riker_PATTERSON295	14-APR-21 05.24.47.000000000 PM
4	50104	Tariq_JORDAN	Tariq_JORDAN353	01-JUN-18 10.14.16.000000000 AM
5	50105	Vito_REYNOLDS	Vito_REYNOLDS496	12-FEB-21 01.25.28.000000000 AM
6	50106	Analee_HAMILTON	Analee_HAMILTON523	23-DEC-18 05.22.23.000000000 AM
7	50107	Audrie_GRAHAM	Audrie_GRAHAM687	30-JAN-18 09.33.37.000000000 AM
8	50110	Devorah_ALEXANDER	Devorah_ALEXANDER406	23-MAR-19 09.35.50.000000000 PM

STEP 2: Create table LOGIN_DETAILS_HISTORY and its corresponding sequence.

NAME	DATATYPE	COLUMN CONSTRAINT	KEY CONSTRAINT
SEQ_ID	NUMBER(10)	NOT NULL	PRIMARY KEY
LOGIN_SEQ_ID	NUMBER(10)		
LOGIN_ID	NUMBER(10)		
USER_NAME	VARCHAR2(50)		
PASSWORD	VARCHAR2(50)		
LOGIN_DATE_TIME	TIMESTAMP(6)		
CREATED_DATE	DATE		
CREATED_BY	VARCHAR2(30)		

```
CREATE SEQUENCE LOGIN_HIST_SEQ
START WITH 1
INCREMENT BY 1
NOCACHE
```

STEP 3: Procedure **LOGIN_DETAILS_BACKUP** is used to archive the records, which moves the records from **LOGIN_DETAILS** table to **LOGIN_DETAILS_HISTORY** table.

```
CREATE OR REPLACE PROCEDURE PROC_LOGIN_DTLS_BACKUP AS

V_CNT               NUMBER(5);
V_FLAG              CHAR(6);
V_FLAG1             CHAR(1);
V_LOGIN_SEQ_ID      NUMBER(10);


CURSOR CUR_REC IS
SELECT * FROM LOGIN_DETAILS
WHERE TRUNC(LOGIN_DATE_TIME) < SYSDATE
AND LOGIN_ID IN (50101,50102,50103,50104,50105,50106,50107,50110);


BEGIN --STEP1

    FOR REC IN CUR_REC
    LOOP
        DBMS_OUTPUT.PUT_LINE('INSIDE LOOP');

        BEGIN --STEP 2

            SELECT COUNT(1)
                INTO V_CNT
            FROM LOGIN_DETAILS_HISTORY
            WHERE LOGIN_ID = REC.LOGIN_ID;

            IF V_CNT = 0 THEN --STEP3
               V_LOGIN_SEQ_ID := 0;
            ELSE
                SELECT LOGIN_SEQ_ID
                    INTO V_LOGIN_SEQ_ID
                FROM LOGIN_DETAILS_HISTORY
                WHERE LOGIN_ID = REC.LOGIN_ID;
            END IF; --STEP3

            DBMS_OUTPUT.PUT_LINE('OLD SEQUENCE ID '|| V_LOGIN_SEQ_ID);

            INSERT INTO LOGIN_DETAILS_HISTORY ( SEQ_ID
                                               ,LOGIN_SEQ_ID
                                               ,LOGIN_ID
```

```
        INSERT INTO LOGIN_DETAILS_HISTORY ( SEQ_ID
                                           ,LOGIN_SEQ_ID
                                           ,LOGIN_ID
                                           ,USER_NAME
                                           ,PASSWORD
                                           ,LOGIN_DATE_TIME
                                           ,CREATED_DATE
                                           ,CREATED_BY)
                                    VALUES (LOGIN_HIST_SEQ.NEXTVAL
                                           ,V_LOGIN_SEQ_ID +1
                                           ,REC.LOGIN_ID
                                           ,REC.USER_NAME
                                           ,REC.PASSWORD
                                           ,REC.LOGIN_DATE_TIME
                                           ,SYSDATE
                                           ,'PROC_LOGIN_DTLS_BACKUP'
                                           );
            V_FLAG := 'TRUE';

        EXCEPTION --STEP2
        WHEN OTHERS THEN
            V_FLAG := 'FALSE';
            DBMS_OUTPUT.PUT_LINE('INSIDE EXCEPTION OF STEP 2');
        END; --STEP2

        BEGIN --STEP 4
            DELETE FROM LOGIN_DETAILS
            WHERE LOGIN_ID = REC.LOGIN_ID;

            V_FLAG1 := 'T';


        EXCEPTION
        WHEN OTHERS THEN
            V_FLAG1 := 'F';
            DBMS_OUTPUT.PUT_LINE('INSIDE EXCEPTION OF STEP 4');
        END; --STEP4

        DBMS_OUTPUT.PUT_LINE('EXISTING LOOP');
    END LOOP;
```

```
    IF V_FLAG = 'TRUE' THEN --STEP5
        IF V_FLAG1 = 'T' THEN --STEP6
            COMMIT;
            DBMS_OUTPUT.PUT_LINE('RECORDS COMMITED');
        ELSE
            ROLLBACK;
            DBMS_OUTPUT.PUT_LINE('RECORDS ROLLBACKED');
        END IF; --STEP6
    ELSE
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('RECORDS ROLLBACKED');
    END IF;--STEP5

END PROC_LOGIN_DTLS_BACKUP; --STEP1
```

STEP 4: After executing the procedure, the records in the **LOGIN_DETAILS_HISTORY** table are as shown below.

```
SELECT * FROM LOGIN_DETAILS_HISTORY
```

Script Output | Query Result

SQL | All Rows Fetched: 9 in 0.027 seconds

	SEQ_ID	LOGIN_SEQ_ID	LOGIN_ID	USER_NAME	PASSWORD	LOGIN_DATE_TIME	CREATED_DATE	CREATED_BY
1	2	1	50100	Kenan_HENDERSON	Kenan_HENDERSON107	01-JUN-19 12.37.27.000000000 AM	30-APR-21	PROC_LOGIN_DTLS_BACKUP
2	3	1	50101	Lyam_COLEMAN	Lyam_COLEMAN803	17-APR-21 05.50.36.000000000 PM	30-APR-21	PROC_LOGIN_DTLS_BACKUP
3	4	1	50102	Phineas_SIMMONS	Phineas_SIMMONS791	04-APR-18 01.34.44.000000000 AM	30-APR-21	PROC_LOGIN_DTLS_BACKUP
4	5	1	50103	Riker_PATTERSON	Riker_PATTERSON295	14-APR-21 05.24.47.000000000 PM	30-APR-21	PROC_LOGIN_DTLS_BACKUP
5	6	1	50104	Tariq_JORDAN	Tariq_JORDAN353	01-JUN-18 10.14.16.000000000 AM	30-APR-21	PROC_LOGIN_DTLS_BACKUP
6	7	1	50105	Vito_REYNOLDS	Vito_REYNOLDS496	12-FEB-21 01.25.28.000000000 AM	30-APR-21	PROC_LOGIN_DTLS_BACKUP
7	8	1	50106	Analee_HAMILTON	Analee_HAMILTON523	23-DEC-18 05.22.23.000000000 AM	30-APR-21	PROC_LOGIN_DTLS_BACKUP
8	9	1	50107	Audrie_GRAHAM	Audrie_GRAHAM687	30-JAN-18 09.33.37.000000000 AM	30-APR-21	PROC_LOGIN_DTLS_BACKUP
9	10	1	50110	Devorah_ALEXANDER	Devorah_ALEXANDER406	23-MAR-19 09.35.50.000000000 PM	30-APR-21	PROC_LOGIN_DTLS_BACKUP

STEP 5: The records in the **LOGIN_DETAILS** table are removed and moved to secondary table as shown below.

```
SELECT * FROM LOGIN_DETAILS
WHERE TRUNC(LOGIN_DATE_TIME) < SYSDATE
AND LOGIN_ID IN (50101,50102,50103,50104,50105,50106,50107,50110);
```

Script Output | Query Result

SQL | All Rows Fetched: 0 in 0.025 seconds

LOGIN_ID	USER_NAME	PASSWORD	LOGIN_D...

FUTURE SCOPE

1. Appointment availability feature can be added via temporary table creation.
2. The database system can be extended to a large-scale clinic management system by adding data on more branches of the clinic.
3. Insurance Beneficiaries:
    - The system caters to the patients irrespective of their dental insurance and the coverage amount.
    - System can be expanded further to the beneficiaries of insurance policy.
4. Financial Accounting:
    - System can take care of the fees that patient pays for the treatment.
    - It can be extended further by managing different levels of salaries and the changes made in the treatment prices over time.
5. Inventory Management:
    - System is assumed to have a single vendor for ordering the products. It can be extended further by introducing multiple vendors and integrating tender management systems.
6. The system can further be extended to provide online diagnostics and consultation.

REFERENCES

https://docs.oracle.com/cd/E11882_01/server.112/e41573/perf_overview.htm#PFGRF025

https://docs.oracle.com/cd/E11882_01/server.112/e40540/indexiot.htm#CNCPT721

https://www.kaggle.com/datasets

https://data.world/