DATABASE PROJECT

BULL CONSTRUCTIONS PVT LTD

Submitted By:

Nikhil Kumar Sonar Chittoju U77486448

Achyuth Reddy Kota U38969276

Venkata Praveen Kumar Dusi U67356284

Contents

1. Introduction

Construction companies use ERP solutions to track the progress, profit & loss, budgeting and staffing control in their projects. This database aims to keep track of the number of clients, projects, staff, equipment, material allocation, payment receivables, wages to staff, material costs in each project.

We have used the Domain Knowledge of Civil Construction Company and past DBA role experiences to design, Develop, Integrate and Test the Database for our Hypothetical Civil construction company i.e., Bull Constructions Pvt Ltd.

Assumptions we have used for Bull Constructions Database:

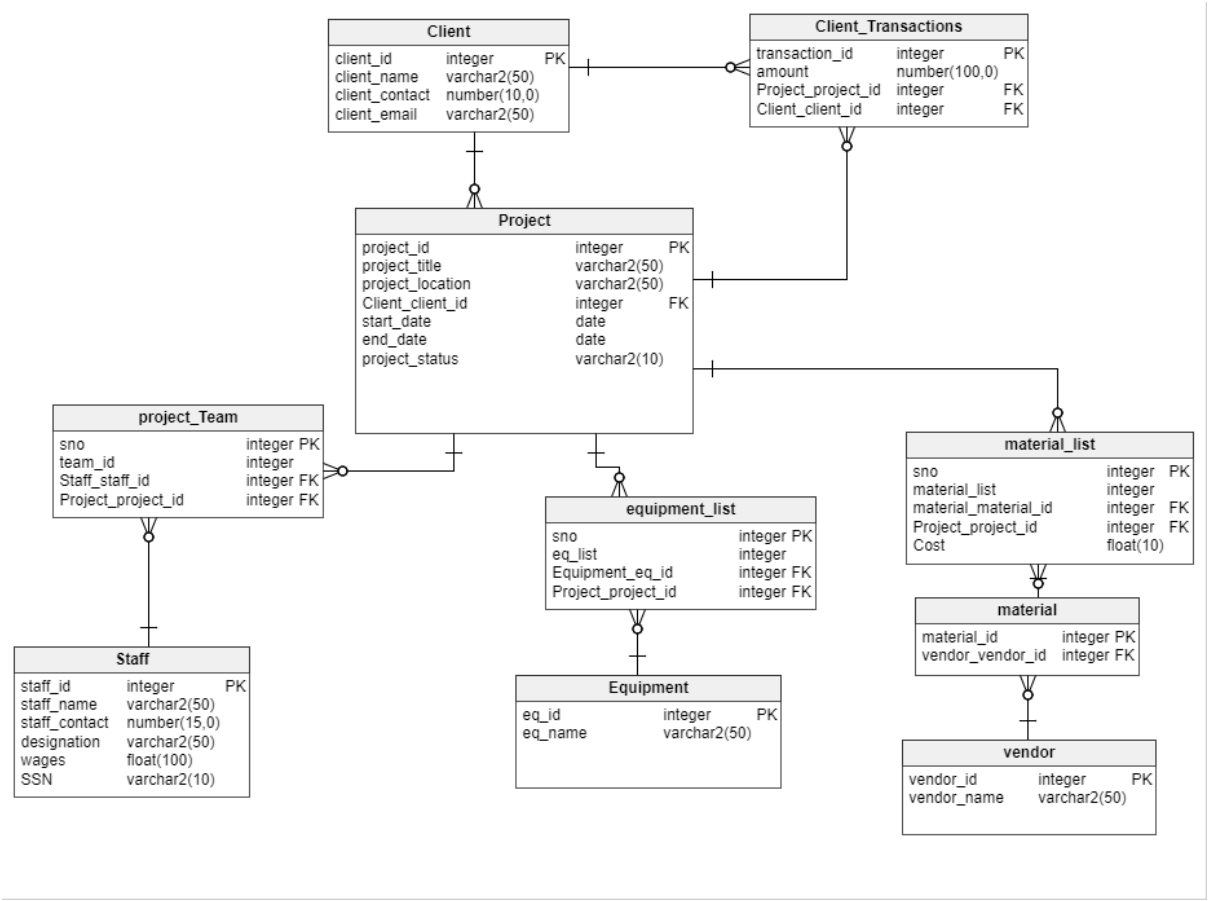
1. Each client can offer many projects to the company.
2. Client pays for the project cost in one installment.
3. Wages of the staff are recorded per project period.
4. Each project has a list of materials and equipment being used.

2. Database Design

This section includes the concept behind creating a database schema, Entity-Relationship diagrams and data dictionaries of Construction company database.

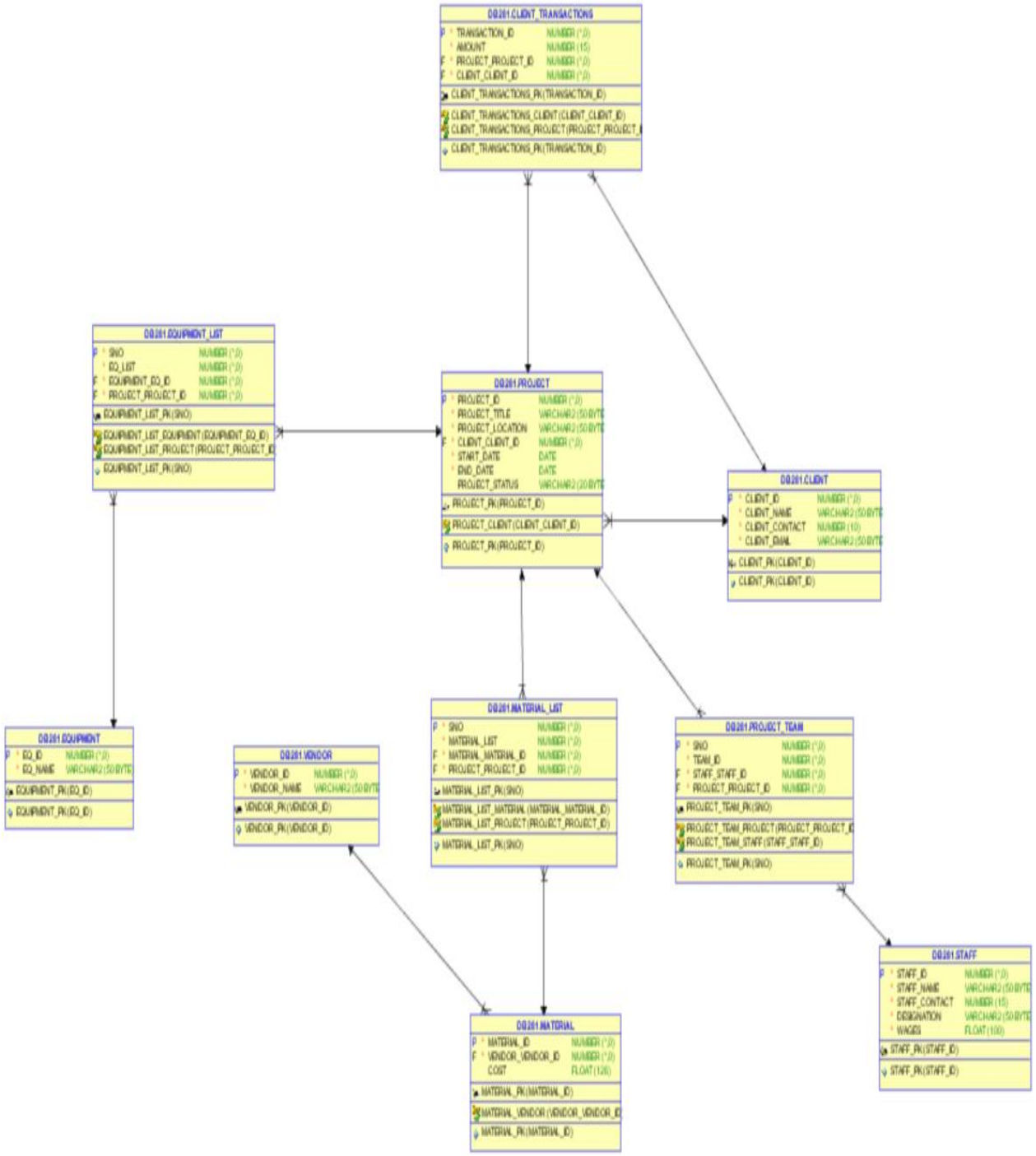
Conceptual Design:

Database design starts with understanding business requirements and mapping those requirements into efficient tables so that we can query relevant information easily. In our case it's a construction company so based on the business requirements and processes we have come up with entities such as client, projects, project teams, staff, equipment, material, vendors and transactions. Mapping all these entities into a schema would result into the picture as shown below.



Logical Design:

Once the requirements are finalized, all the tables and their associated views, indexes, constraints, dependencies etc. are created based on the conceptual design in Oracle SQL Developer. After creating the necessary tables as per the conceptual design, the logical design is automatically generated from import > data dictionary option, in the file menu's Data Modeler option.



3. Physical Database Design

Physical Database design refers to the conversion of relations in the logical data base into corresponding database objects. We are using the SQL Developer for the creation and development of the database objects based on the logical model developed. All the requirements are gathered based on the business and logical model is developed. Physical design involved in creating various database objects like tables, indexes and views etc. based on entities and relations in logical design.

3.1. Table Project

Description: This table holds project details of the company

3.1.1. Columns

Column name	Type	Properties	Description
project_id	Integer	PK	
project_title	varchar2(50)		
project_location	varchar2(50)		
Client_client_id	Integer		
start_date	Date		
end_date	Date		
project_status	varchar2(10)		

3.1.2 Data Dictionary

```
-- Table: Project
CREATE TABLE Project (
    project_id integer  NOT NULL,
    project_title varchar2(50)  NOT NULL,
    project_location varchar2(50)  NOT NULL,
    Client_client_id integer  NOT NULL,
    start_date date  NOT NULL,
    end_date date  NOT NULL,
    project_status varchar2(10)  NOT NULL,
    CONSTRAINT Project_pk PRIMARY KEY (project_id)
) ;
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PROJECT_ID	NUMBER(38,0)	No	(null)	1	(null)
2	PROJECT_TITLE	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	PROJECT_LOCATION	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4	CLIENT_CLIENT_ID	NUMBER(38,0)	No	(null)	4	(null)
5	START_DATE	DATE	No	(null)	5	(null)
6	END_DATE	DATE	No	(null)	6	(null)
7	PROJECT_STATUS	VARCHAR2(20 BYTE)	Yes	(null)	7	(null)

3.1.3 Constraints

```
-- Reference: Project_Client (table: Project)
ALTER TABLE Project ADD CONSTRAINT Project_Client
    FOREIGN KEY (Client_client_id)
    REFERENCES Client (client_id);
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE
1	PROJECT_CLIENT	Foreign_Key	(null)	DB281	CLIENT	CLIENT_PK	NO ACTION	ENABLED	NOT DEFERRABLE
2	PROJECT_PK	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
3	SYS_C0069338	Check	"PROJECT_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
4	SYS_C0069339	Check	"PROJECT_TITLE" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
5	SYS_C0069340	Check	"PROJECT_LOCATION" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
6	SYS_C0069341	Check	"CLIENT_CLIENT_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
7	SYS_C0069342	Check	"START_DATE" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
8	SYS_C0069343	Check	"END_DATE" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE

3.1.4 Dependencies

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	OWNER	NAME	TYPE	REFERENCED_OWNER	REFERENCED_NAME	REFERENCED_TYPE
1	DB281	COMP_PROFIT	PROCEDURE	DB281	PROJECT	TABLE

3.1.5 Indexes

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	DB281	END_DATE	NONUNIQUE	VALID	NORMAL	N	NO	(null)	NO	END_DATE
2	DB281	PROJECT_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	PROJECT_ID
3	DB281	PROJECT_TITLE_IDX	NONUNIQUE	VALID	NORMAL	N	NO	(null)	NO	PROJECT_TITLE

3.2. Table Client

Description: This table holds the Data of the Clients

3.2.1. Columns

Column name	Type	Properties	Description
client_id	integer	PK	
client_name	varchar2(50)		
client_contact	number(10,0)		
client_email	varchar2(50)		

3.2.2 Data Dictionary

```
-- Table: Client
CREATE TABLE Client (
    client_id integer  NOT NULL,
    client_name varchar2(50)  NOT NULL,
    client_contact number(10,0)  NOT NULL,
    client_email varchar2(50)  NOT NULL,
    CONSTRAINT Client_pk PRIMARY KEY (client_id)
) ;
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CLIENT_ID	NUMBER(38,0)	No	(null)	1	(null)
2	CLIENT_NAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	CLIENT_CONTACT	NUMBER(10,0)	No	(null)	3	(null)
4	CLIENT_EMAIL	VARCHAR2(50 BYTE)	No	(null)	4	(null)

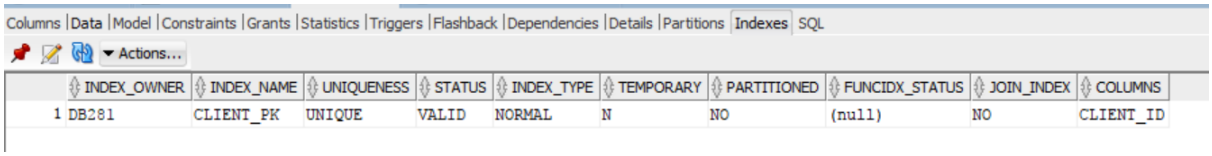
3.2.3 Constraints

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE	VALIDATED
1	CLIENT_PK	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED
2	SYS_C0069325	Check	"CLIENT_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED
3	SYS_C0069326	Check	"CLIENT_NAME" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED
4	SYS_C0069327	Check	"CLIENT_CONTACT" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED
5	SYS_C0069328	Check	"CLIENT_EMAIL" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED

3.2.4 Indexes



	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	DB281	CLIENT_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	CLIENT_ID

3.3. Table Client Transactions

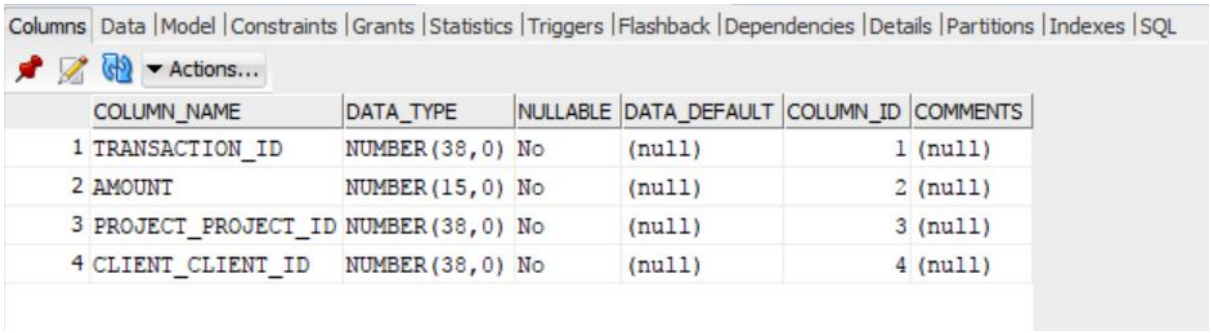
Description: This table holds the Transactions between company and clients

3.3.1. Columns

Column name	Type	Properties	Description
transaction_id	Integer	PK	
Amount	number(100,0)		
Project_project_id	Integer		
Client_client_id	Integer		

3.3.2 Data Dictionary

```
CREATE TABLE Client_Transactions (
    transaction_id integer  NOT NULL,
    amount number(100,0)  NOT NULL,
    Project_project_id integer  NOT NULL,
    Client_client_id integer  NOT NULL,
    CONSTRAINT Client_Transactions_pk PRIMARY KEY (transaction_id)
) ;
```



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	TRANSACTION_ID	NUMBER(38,0)	No	(null)	1	(null)
2	AMOUNT	NUMBER(15,0)	No	(null)	2	(null)
3	PROJECT_PROJECT_ID	NUMBER(38,0)	No	(null)	3	(null)
4	CLIENT_CLIENT_ID	NUMBER(38,0)	No	(null)	4	(null)

3.3.3 Constraints

```
-- Reference: Client_Transactions_Client (table: Client_Transactions)
ALTER TABLE Client_Transactions ADD CONSTRAINT Client_Transactions_Client
    FOREIGN KEY (Client_client_id)
    REFERENCES Client (client_id);

-- Reference: Client_Transactions_Project (table: Client_Transactions)
ALTER TABLE Client_Transactions ADD CONSTRAINT Client_Transactions_Project
    FOREIGN KEY (Project_project_id)
    REFERENCES Project (project_id);
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE
1	CLIENT_TRANSACTIONS_CLIENT	Foreign_Key	(null)	DB281	CLIENT	CLIENT_PK	NO ACTION	ENABLED	NOT DEFERRABLE
2	CLIENT_TRANSACTIONS_PK	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
3	CLIENT_TRANSACTIONS_PROJECT	Foreign_Key	(null)	DB281	PROJECT	PROJECT_PK	NO ACTION	ENABLED	NOT DEFERRABLE
4	SYS_C0069330	Check	"TRANSACTION_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
5	SYS_C0069331	Check	"AMOUNT" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
6	SYS_C0069332	Check	"PROJECT_PROJECT_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
7	SYS_C0069333	Check	"CLIENT_CLIENT_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE

3.3.3 Dependencies

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	OWNER	NAME	TYPE	REFERENCED_OWNER	REFERENCED_NAME	REFERENCED_TYPE
1	DB281	AVG_PROFIT	FUNCTION	DB281	CLIENT_TRANSACTIONS	TABLE
2	DB281	COMP_PROFIT	PROCEDURE	DB281	CLIENT_TRANSACTIONS	TABLE

3.3.4 Indexes

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	DB281	CLIENT_TRANSACTIONS_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	TRANSACTION_ID

3.4. Table Staff

Description: This table holds the details of the staff of the company

3.4.1. Columns

Column name	Type	Properties	Description
staff_id	integer	PK	
staff_name	varchar2(50)		
staff_contact	number (15,0)		
designation	varchar2(50)		
wages	float (100)		

3.4.2 Data Dictionary

```
-- Table: Staff
CREATE TABLE Staff (
    staff_id integer  NOT NULL,
    staff_name varchar2(50)  NOT NULL,
    staff_contact number(15,0)  NOT NULL,
    designation varchar2(50)  NOT NULL,
    wages float(100)  NOT NULL,
    CONSTRAINT Staff_pk PRIMARY KEY (staff_id)
) ;
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	STAFF_ID	NUMBER(38,0)	No	(null)	1	(null)
2	STAFF_NAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	STAFF_CONTACT	NUMBER(15,0)	No	(null)	3	(null)
4	DESIGNATION	VARCHAR2(50 BYTE)	No	(null)	4	(null)
5	WAGES	FLOAT	No	(null)	5	(null)
6	SSN	VARCHAR2(20 BYTE)	Yes	(null)	6	(null)

3.4.2 Constraints

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE
1	STAFF_PK	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
2	SYS_C0069345	Check	"STAFF_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
3	SYS_C0069346	Check	"STAFF_NAME" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
4	SYS_C0069347	Check	"STAFF_CONTACT" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
5	SYS_C0069348	Check	"DESIGNATION" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
6	SYS_C0069349	Check	"WAGES" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE

3.4.3 Dependencies

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	OWNER	NAME	TYPE	REFERENCED_OWNER	REFERENCED_NAME	REFERENCED_TYPE
1	DB281	AVG_PROFIT	FUNCTION	DB281	STAFF	TABLE
2	DB281	COMP_PROFIT	PROCEDURE	DB281	STAFF	TABLE

3.4.4 Indexes

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	DB281	STAFF_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	STAFF_ID

3.5. Table project_Team

Description: This table holds the Details of the project team assigned to the project

3.5.1. Columns

Column name	Type	Properties	Description
Sno	integer	PK	
team_id	integer		
Staff_staff_id	integer		
Project_project_id	integer		

3.5.2 Data Dictionary

```
-- Table: project_Team
CREATE TABLE project_Team (
    sno integer  NOT NULL,
    team_id integer  NOT NULL,
    Staff_staff_id integer  NOT NULL,
    Project_project_id integer  NOT NULL,
    CONSTRAINT project_Team_pk PRIMARY KEY (sno)
) ;
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SNO	NUMBER(38,0)	No	(null)	1	(null)
2	TEAM_ID	NUMBER(38,0)	No	(null)	2	(null)
3	STAFF_STAFF_ID	NUMBER(38,0)	No	(null)	3	(null)
4	PROJECT_PROJECT_ID	NUMBER(38,0)	No	(null)	4	(null)

3.5.2 Constraints

```
-- Reference: project_Team_Project (table: project_Team)
ALTER TABLE project_Team ADD CONSTRAINT project_Team_Project
    FOREIGN KEY (Project_project_id)
    REFERENCES Project (project_id);

-- Reference: project_Team_Staff (table: project_Team)
ALTER TABLE project_Team ADD CONSTRAINT project_Team_Staff
    FOREIGN KEY (Staff_staff_id)
    REFERENCES Staff (staff_id);
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE
1	PROJECT_TEAM_PK	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
2	PROJECT_TEAM_PROJECT	Foreign_Key	(null)	DB281	PROJECT	PROJECT_PK	NO ACTION	ENABLED	NOT DEFERRABLE
3	PROJECT_TEAM_STAFF	Foreign_Key	(null)	DB281	STAFF	STAFF_PK	NO ACTION	ENABLED	NOT DEFERRABLE
4	SYS_C0069364	Check	"SNO" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
5	SYS_C0069365	Check	"TEAM_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
6	SYS_C0069366	Check	"STAFF_STAFF_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
7	SYS_C0069367	Check	"PROJECT_PROJECT_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE

3.5.3 Dependencies

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	OWNER	NAME	TYPE	REFERENCED_OWNER	REFERENCED_NAME	REFERENCED_TYPE
1	DB281	COMP_PROFIT	PROCEDURE	DB281	PROJECT_TEAM	TABLE

3.5.4 Indexes

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	DB281	PROJECT_TEAM_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	SNO

3.6. Table Equipment

Description: This holds the details of the equipment of the company

3.6.1. Columns

Column name	Type	Properties	Description
eq_id	integer	PK	
eq_name	varchar2(50)		

3.6.2 Data Dictionary

```
-- Table: Equipment
CREATE TABLE Equipment (
    eq_id integer  NOT NULL,
    eq_name varchar2(50)  NOT NULL,
    CONSTRAINT Equipment_pk PRIMARY KEY (eq_id)
) ;
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EQ_ID	NUMBER(38,0)	No	(null)	1	(null)
2	EQ_NAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)

3.6.2 Constraints

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE	VALIDATED
1	EQUIPMENT_PK	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED
2	SYS_C0069335	Check	"EQ_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED
3	SYS_C0069336	Check	"EQ_NAME" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED

3.6.3 Indexes

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	DB281	EQUIPMENT_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	EQ_ID

3.7. Table vendor

Description: This contains the list of vendors for the project

3.7.1. Columns

Column name	Type	Properties	Description
vendor_id	integer	PK	
vendor_name	varchar2(50)		

3.7.2 Data Dictionary

```
-- Table: vendor
CREATE TABLE vendor (
    vendor_id integer  NOT NULL,
    vendor_name varchar2(50)  NOT NULL,
    CONSTRAINT vendor_pk PRIMARY KEY (vendor_id)
) ;
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	VENDOR_ID	NUMBER(38,0)	No	(null)	1	(null)
2	VENDOR_NAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)

3.7.3 Constraints

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE	VALIDATED
1	SYS_C0069369	Check	"VENDOR_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED
2	SYS_C0069370	Check	"VENDOR_NAME" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED
3	VENDOR_PK	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE	VALIDATED

3.7.4 Indexes

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	DB281	VENDOR_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	VENDOR_ID

2.8. Table material

Description: This table contains the materials used by the company

3.8.1. Columns

Column name	Type	Properties	Description
material_id	integer	PK	
vendor_vendor_id	integer		

3.8.2 Data Dictionary

```
-- Table: material
CREATE TABLE material (
    material_id integer  NOT NULL,
    vendor_vendor_id integer  NOT NULL,
    CONSTRAINT material_pk PRIMARY KEY (material_id)
) ;
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	VENDOR_ID	NUMBER(38,0)	No	(null)	1	(null)
2	VENDOR_NAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)

3.8.3 Constraints

```
-- Reference: material_vendor (table: material)
ALTER TABLE material ADD CONSTRAINT material_vendor
    FOREIGN KEY (vendor_vendor_id)
    REFERENCES vendor (vendor_id);
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS	DEFERRABLE
1	SYS_C0069369	Check	"VENDOR_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
2	SYS_C0069370	Check	"VENDOR_NAME" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE
3	VENDOR_PK	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED	NOT DEFERRABLE

3.8.4 Indexes

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	DB281	VENDOR_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	VENDOR_ID

3.9. Table equipment_list

Description: This table holds the list of equipment used in a project

3.9.1. Columns

Column name	Type	Properties	Description
sno	integer	PK	
eq_list	integer		
Equipment_eq_id	integer		
Project_project_id	integer		

3.9.2 Data Dictionary

```
-- Table: equipment_list
CREATE TABLE equipment_list (
    sno integer  NOT NULL,
    eq_list integer  NOT NULL,
    Equipment_eq_id integer  NOT NULL,
    Project_project_id integer  NOT NULL,
    CONSTRAINT equipment_list_pk PRIMARY KEY (sno)
) ;
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SNO	NUMBER(38,0)	No	(null)	1	(null)
2	EQ_LIST	NUMBER(38,0)	No	(null)	2	(null)
3	EQUIPMENT_EQ_ID	NUMBER(38,0)	No	(null)	3	(null)
4	PROJECT_PROJECT_ID	NUMBER(38,0)	No	(null)	4	(null)

3.9.3 Constraints

```
-- Reference: equipment_list_Equipment (table: equipment_list)
ALTER TABLE equipment_list ADD CONSTRAINT equipment_list_Equipment
    FOREIGN KEY (Equipment_eq_id)
    REFERENCES Equipment (eq_id);

-- Reference: equipment_list_Project (table: equipment_list)
ALTER TABLE equipment_list ADD CONSTRAINT equipment_list_Project
    FOREIGN KEY (Project_project_id)
    REFERENCES Project (projec  t_id);
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS
1	EQUIPMENT_LIST_EQUIPMENT	Foreign_Key	(null)	DB281	EQUIPMENT	EQUIPMENT_PK	NO ACTION	ENABLED
2	EQUIPMENT_LIST_PK	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED
3	EQUIPMENT_LIST_PROJECT	Foreign_Key	(null)	DB281	PROJECT	PROJECT_PK	NO ACTION	ENABLED
4	SYS_C0069351	Check	"SNO" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED
5	SYS_C0069352	Check	"EQ_LIST" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED
6	SYS_C0069353	Check	"EQUIPMENT_EQ_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED
7	SYS_C0069354	Check	"PROJECT_PROJECT_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED

3.9.4 Indexes

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	DB281	EQUIPMENT_LIST_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	SNO

3.10. Table material_list

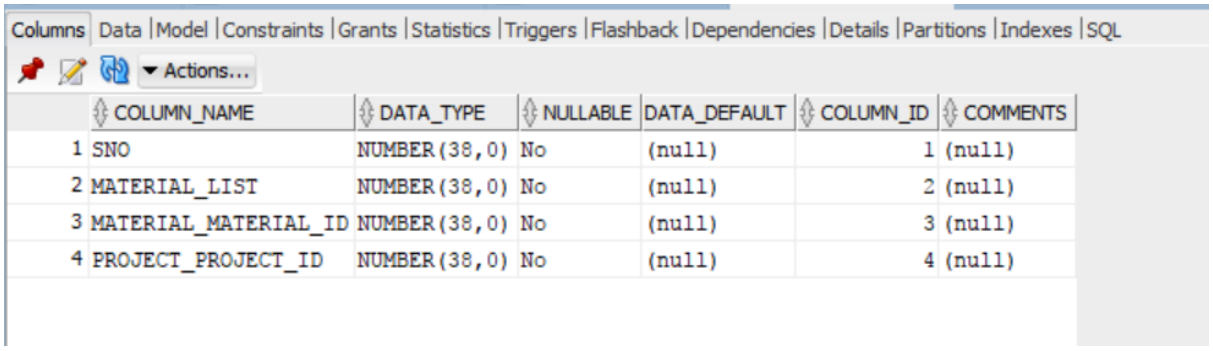
Description: This contains the List of materials used for a product

3.10.1. Columns

Column name	Type	Properties	Description
Sno	integer	PK	
material_list	integer		
material_material_id	integer		
Project_project_id	integer		
Cost	float (10)		

3.10.2 Data Dictionary

```
-- Table: material_list
CREATE TABLE material_list (
    sno integer  NOT NULL,
    material_list integer  NOT NULL,
    material_material_id integer  NOT NULL,
    Project_project_id integer  NOT NULL,
    Cost float(10)  NOT NULL,
    CONSTRAINT material_list_pk PRIMARY KEY (sno)
) ;
```



Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SNO	NUMBER(38,0)	No	(null)	1	(null)
2	MATERIAL_LIST	NUMBER(38,0)	No	(null)	2	(null)
3	MATERIAL_MATERIAL_ID	NUMBER(38,0)	No	(null)	3	(null)
4	PROJECT_PROJECT_ID	NUMBER(38,0)	No	(null)	4	(null)

3.10.3 Constraints

```
-- Reference: material_list_Project (table: material_list)
ALTER TABLE material_list ADD CONSTRAINT material_list_Project
    FOREIGN KEY (Project_project_id)
    REFERENCES Project (project_id);

-- Reference: material_list_material (table: material_list)
ALTER TABLE material_list ADD CONSTRAINT material_list_material
    FOREIGN KEY (material_material_id)
    REFERENCES material (material_id);
```

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATUS
1	MATERIAL_LIST_MATERIAL	Foreign_Key	(null)	DB281	MATERIAL	MATERIAL_PK	NO ACTION	ENABLED
2	MATERIAL_LIST_PK	Primary_Key	(null)	(null)	(null)	(null)	(null)	ENABLED
3	MATERIAL_LIST_PROJECT	Foreign_Key	(null)	DB281	PROJECT	PROJECT_PK	NO ACTION	ENABLED
4	SYS_C0069359	Check	"SNO" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED
5	SYS_C0069360	Check	"MATERIAL_LIST" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED
6	SYS_C0069361	Check	"MATERIAL_MATERIAL_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED
7	SYS_C0069362	Check	"PROJECT_PROJECT_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLED

3.10.4 Indexes

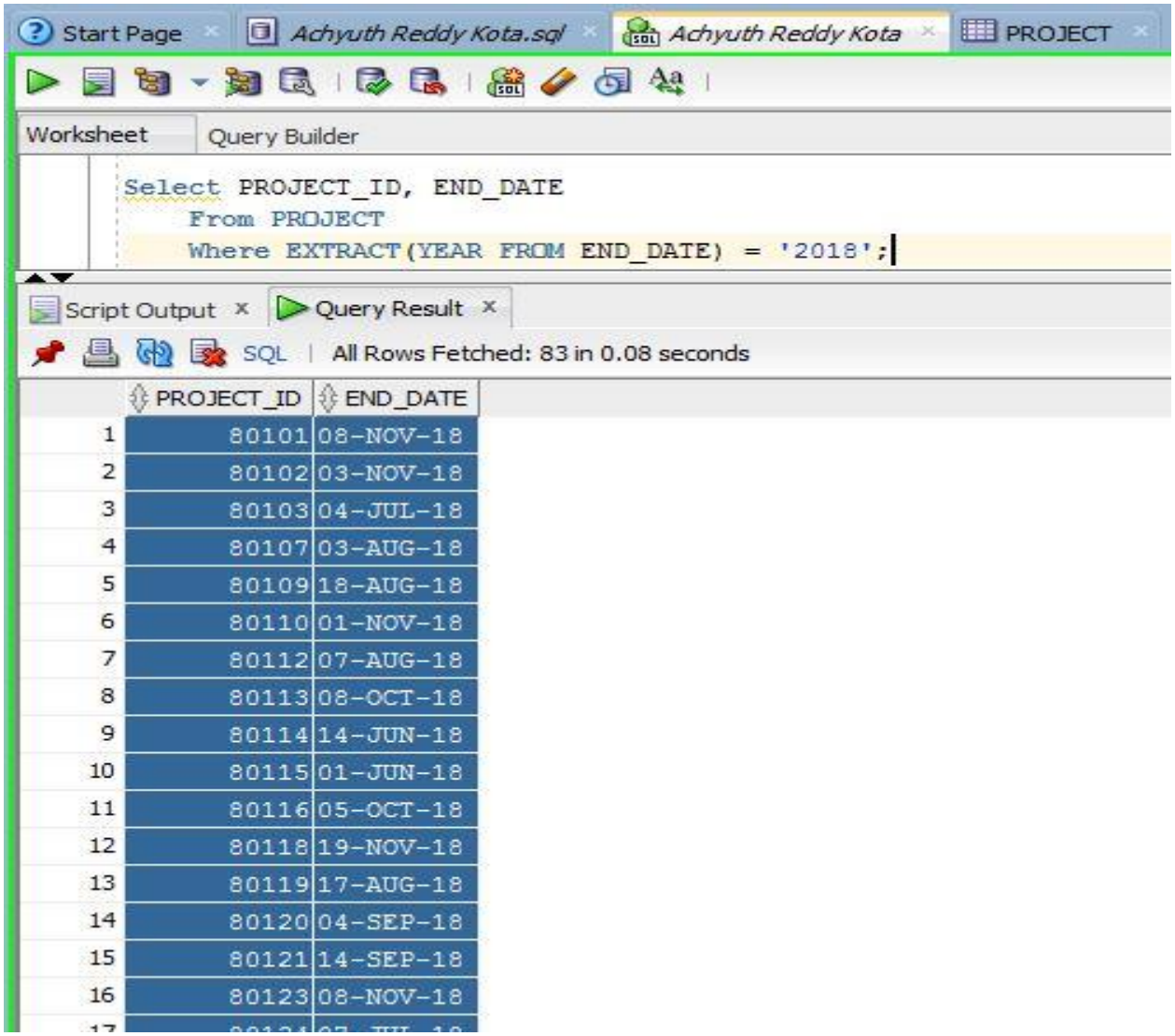
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS
1	DB281	MATERIAL_LIST_PK	UNIQUE	VALID	NORMAL	N	NO	(null)	NO	SNO
2	DB281	INDEX_MATERIAL_LIST	NONUNIQUE	VALID	NORMAL	N	NO	(null)	NO	MATERIAL_LIST
3	DB281	INDEX_PROJECT_PROJECT_ID	NONUNIQUE	VALID	NORMAL	N	NO	(null)	NO	PROJECT_PROJECT_ID

4. Query Writing

- **Interesting Queries:**
  - **The company is planning the estimates for next year. They are curious to know the projects that are ending in the year "2018"**



- **The company wants to Consider the salaries spent based on designations.**

```
1  select designation,sum(wages) from staff s
2  group by designation;
3
```

Results:

```
11  select designation,sum(wages) from staff
12  group by designation order by sum(wages)desc;
```

Query Result

SQL | All Rows Fetched: 8 in 0.05 seconds

	DESIGNATION	SUM(WAGES)
1	Construction Manager	533622.92
2	Accountant	524706.62
3	Sr. Construction Manager	523180.82
4	Project Manager	520445.64
5	Sr. Engineer	400000
6	Engineer	300000
7	Supervisor	200000
8	Operator	100000

- After considering the salaries based on the designations the company wants to see the spending on the Projects on each designation.

```
select pt.PROJECT_PROJECT_ID, designation,sum(wages) from staff s inner join project_team pt
on s.staff_id=pt.STAFF_STAFF_ID
group by pt.PROJECT_PROJECT_ID,s.DESIGNATION
order by sum(wages) desc;
```

Results:

```
select pt.PROJECT_PROJECT_ID, designation,sum(wages) from staff s inner join project_team pt
on s.staff_id=pt.STAFF_STAFF_ID
group by pt.PROJECT_PROJECT_ID,s.DESIGNATION
order by sum(wages) desc;
```

Query Result

SQL | Fetched 50 rows in 0.038 seconds

	PROJECT_PROJECT_ID	DESIGNATION	SUM(WAGES)
1	80144	Construction Manager	113660.56
2	80144	Sr. Construction Manager	107390.38
3	80144	Accountant	106303.67
4	80144	Project Manager	103956.99
5	80144	Sr. Engineer	84000
6	80137	Accountant	67730.66
7	80137	Construction Manager	64520.99
8	80144	Engineer	63000
9	80137	Sr. Construction Manager	50099.45

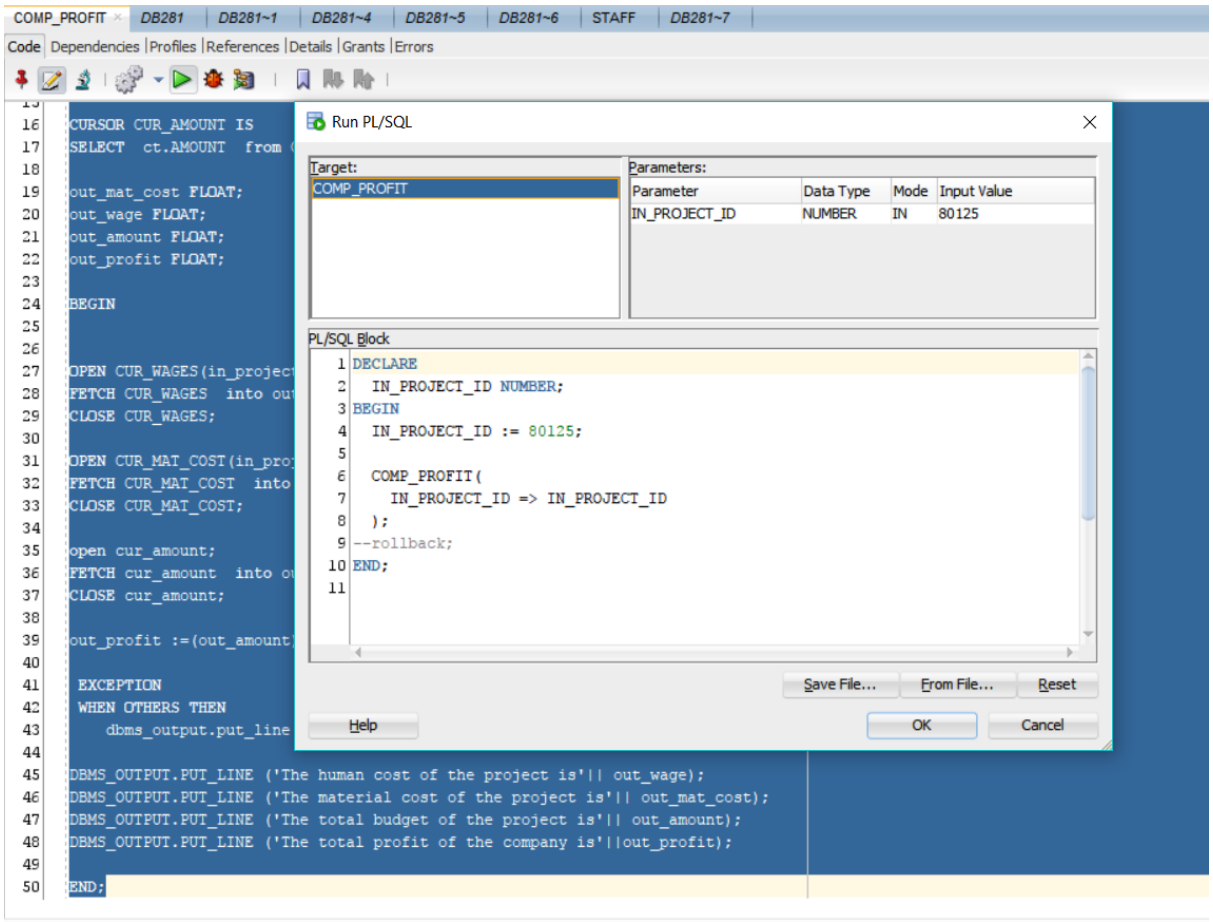
5. Stored Procedure Programming

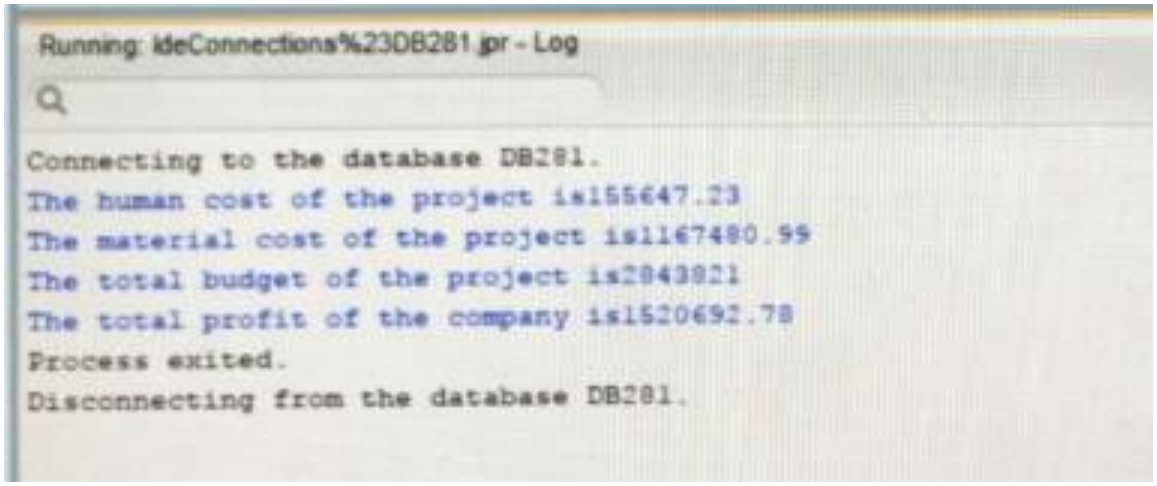
➢ Exploring Stored Procedures:

In the stored Procedure Programming, we came up with a functionality to consider the profitability of a Project proposal. In this program the company can estimate the profits on a project. This would be useful for them to consider the prospects of taking that project.

```
create or replace PROCEDURE COMP_PROFIT
(in_project_id       project.project_id%type)
AS

CURSOR CUR_WAGES(in_project_id project.project_id%TYPE) IS
SELECT sum( distinct st.wages) from  staff st, project_team pt
WHERE st.STAFF_ID=pt.STAFF_STAFF_ID
AND pt.PROJECT_PROJECT_ID=in_project_id;

CURSOR CUR_MAT_COST(in_project_id project.project_id%TYPE) IS
SELECT sum(distinct m.COST) from  MATERIAL_LIST ml ,project p,MATERIAL m
WHERE ml.PROJECT_PROJECT_ID=p.PROJECT_ID
AND ml.MATERIAL_MATERIAL_ID=m.MATERIAL_ID
AND p.project_id=in_project_id;

CURSOR CUR_AMOUNT IS
SELECT  ct.AMOUNT  from CLIENT_TRANSACTIONS ct  where  project_PROJECT_ID=in_project_id;

out_mat_cost FLOAT;
out_wage FLOAT;
out_amount FLOAT;
out_profit FLOAT;

BEGIN


OPEN CUR_WAGES(in_project_id);
FETCH CUR_WAGES  into out_wage;
CLOSE CUR_WAGES;

OPEN CUR_MAT_COST(in_project_id);
FETCH CUR_MAT_COST  into out_mat_cost;
CLOSE CUR_MAT_COST;

open cur_amount;
FETCH cur_amount  into out_amount;
CLOSE cur_amount;

out_profit :=(out_amount)-(out_mat_cost+out_wage);

 EXCEPTION
 WHEN OTHERS THEN
    dbms_output.put_line ('Error while calculsting the  ');

DBMS_OUTPUT.PUT_LINE ('The human cost of the project is'|| out_wage);
DBMS_OUTPUT.PUT_LINE ('The material cost of the project is'|| out_mat_cost);
DBMS_OUTPUT.PUT_LINE ('The total budget of the project is'|| out_amount);
DBMS_OUTPUT.PUT_LINE ('The total profit of the company is'||out_profit);

END;
```

Results:


COMP_PROFIT
DB281
DB281~1
DB281~4
DB281~5
DB281~6
STAFF
DB281~7
Code
Dependencies
Profiles
References
Details
Grants
Errors
Run PL/SQL
Target:
COMP_PROFIT
Parameters:
Parameter
Data Type
Mode
Input Value
IN_PROJECT_ID
NUMBER
IN
80125
PL/SQL Block
DECLARE
IN_PROJECT_ID NUMBER;
BEGIN
IN_PROJECT_ID := 80125;
COMP_PROFIT(
IN_PROJECT_ID => IN_PROJECT_ID
);
--rollback;
END;
Save File...
From File...
Reset
Help
OK
Cancel
CURSOR CUR_AMOUNT IS
SELECT ct.AMOUNT from
out_mat_cost FLOAT;
out_wage FLOAT;
out_amount FLOAT;
out_profit FLOAT;
BEGIN
OPEN CUR_WAGES(in_project
FETCH CUR_WAGES into ou
CLOSE CUR_WAGES;
OPEN CUR_MAT_COST(in_proj
FETCH CUR_MAT_COST into
CLOSE CUR_MAT_COST;
open cur_amount;
FETCH cur_amount into ou
CLOSE cur_amount;
out_profit :=(out_amount
EXCEPTION
WHEN OTHERS THEN
dbms_output.put_line
DBMS_OUTPUT.PUT_LINE ('The human cost of the project is'|| out_wage);
DBMS_OUTPUT.PUT_LINE ('The material cost of the project is'|| out_mat_cost);
DBMS_OUTPUT.PUT_LINE ('The total budget of the project is'|| out_amount);
DBMS_OUTPUT.PUT_LINE ('The total profit of the company is'||out_profit);
END;


Running: IdeConnections%23DB281.jpr - Log
Connecting to the database DB281.
The human cost of the project is155647.23
The material cost of the project is1167480.99
The total budget of the project is2843821
The total profit of the company is1520692.78
Process exited.
Disconnecting from the database DB281.

➢ Exploring Functions

The company wants the overall average profit. We have used functions to compute the average profit using Custom Functions in SQL

```
create or replace FUNCTION AVG_PROFIT

RETURN float IS
   avg_profit float := 0;

out_avg_wage   FLOAT;
out_avg_mat    FLOAT;
out_avg_amount FLOAT;

cursor ave_wage IS select avg(wages) from STAFF;
cursor avg_cost  is select avg(cost)   from MATERIAL;
cursor  avg_amount is select avg(amount) from CLIENT_TRANSACTIONS;

BEGIN
open ave_wage;
fetch ave_wage into out_avg_wage;
close ave_wage;

open avg_cost;
fetch avg_cost into out_avg_mat;
close avg_cost;

open avg_amount;
fetch avg_amount into out_avg_amount;
avg_profit :=out_avg_amount-(out_avg_wage+out_avg_amount);
END;
```

6. Performance Tuning

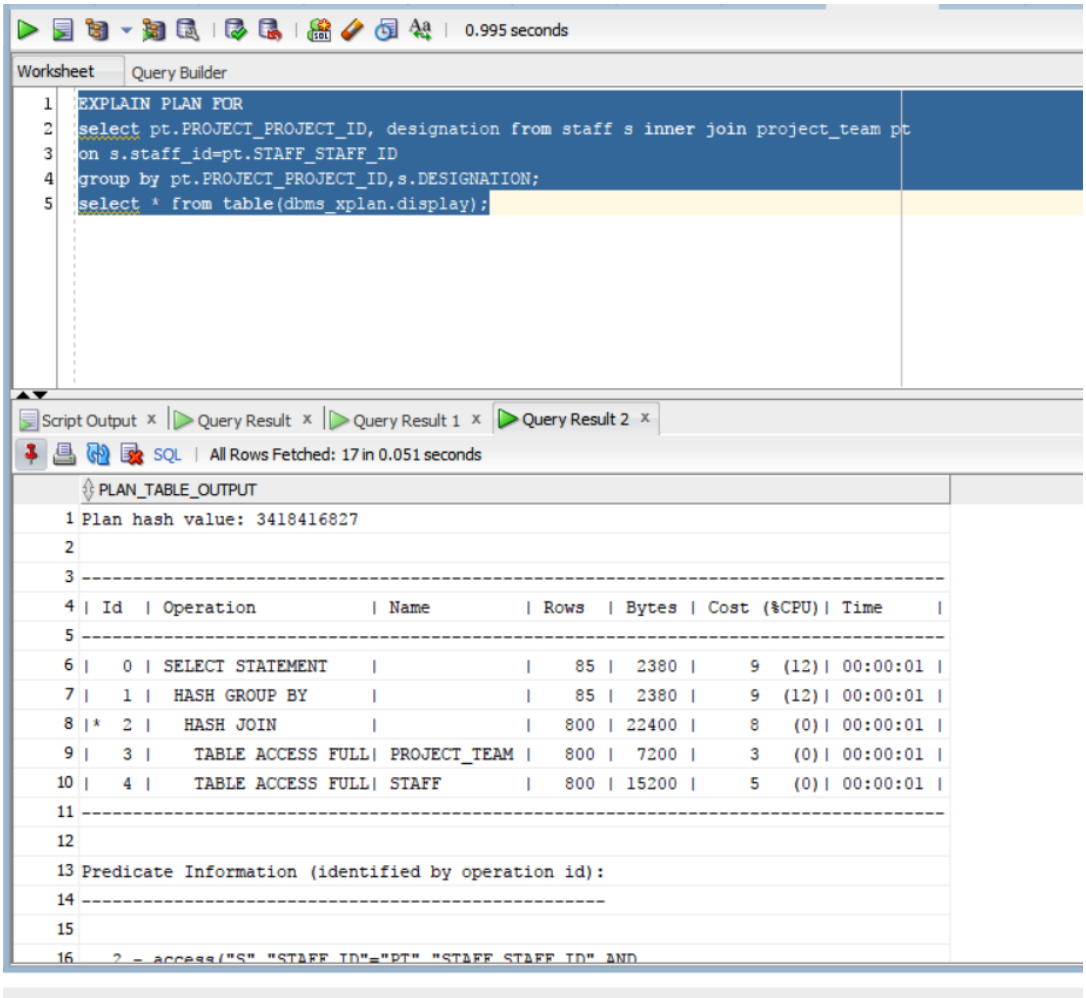
➢ Indexing:

In this section we have tuned the queries in Chapter 5 by adding indexes

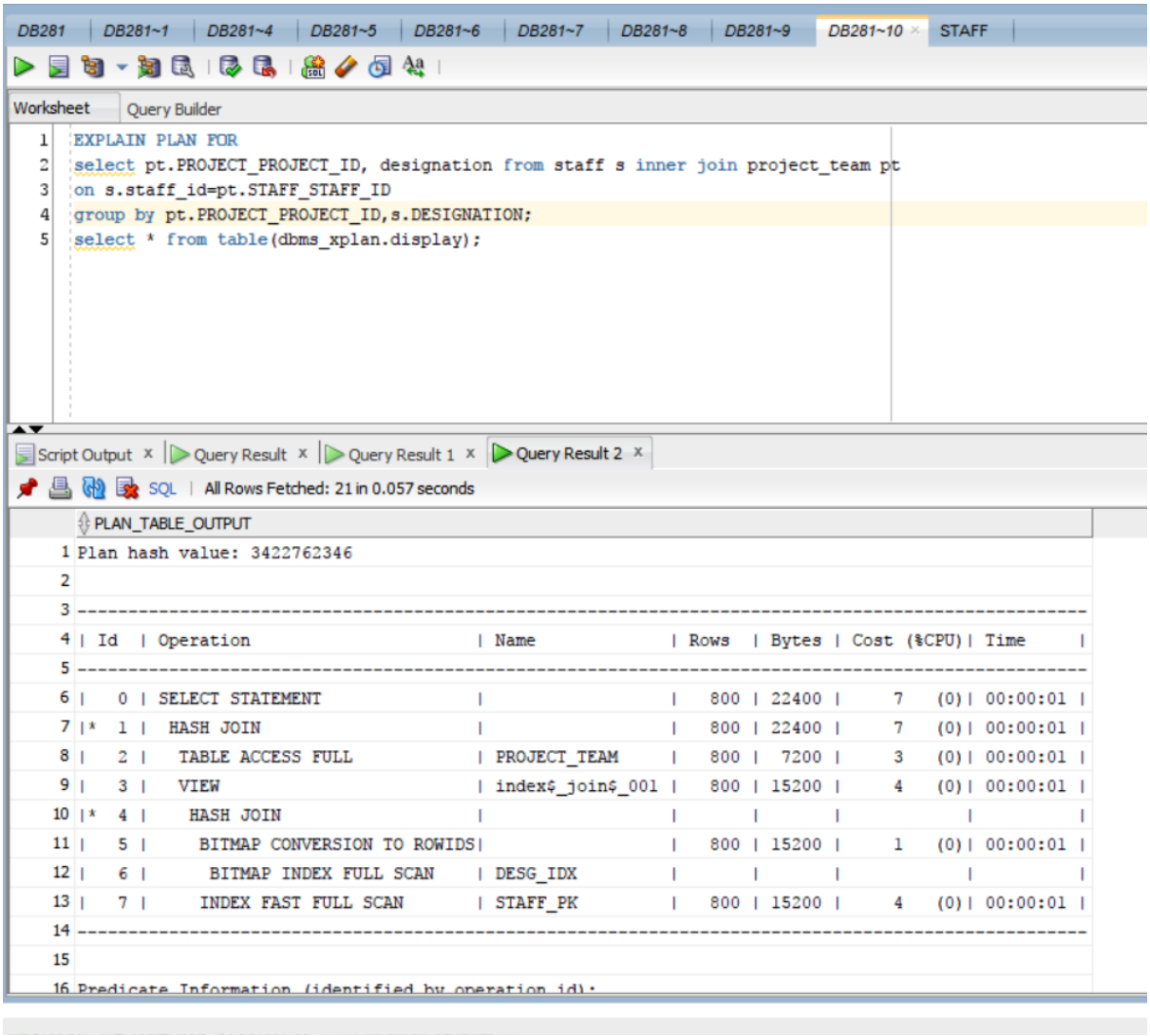
EXPLAIN PLAN:

```
1  EXPLAIN PLAN FOR
2  select pt.PROJECT_PROJECT_ID, designation from staff s inner join project_team pt
3  on s.staff_id=pt.STAFF_STAFF_ID
4  group by pt.PROJECT_PROJECT_ID,s.DESIGNATION;
5  select * from table(dbms_xplan.display);
```

Without Tuning:



With Tuning:


DB281
DB281~1
DB281~4
DB281~5
DB281~6
DB281~7
DB281~8
DB281~9
DB281~10
STAFF
Worksheet
Query Builder
1 EXPLAIN PLAN FOR
2 select pt.PROJECT_PROJECT_ID, designation from staff s inner join project_team pt
3 on s.staff_id=pt.STAFF_STAFF_ID
4 group by pt.PROJECT_PROJECT_ID,s.DESIGNATION;
5 select * from table(dbms_xplan.display);
Script Output
Query Result
Query Result 1
Query Result 2
SQL
All Rows Fetched: 21 in 0.057 seconds
PLAN_TABLE_OUTPUT
1 Plan hash value: 3422762346
2
3 ----------
4 | Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |
5 ----------
6 | 0 | SELECT STATEMENT | | 800 | 22400 | 7 (0)| 00:00:01 |
7 |* 1 | HASH JOIN | | 800 | 22400 | 7 (0)| 00:00:01 |
8 | 2 | TABLE ACCESS FULL | PROJECT_TEAM | 800 | 7200 | 3 (0)| 00:00:01 |
9 | 3 | VIEW | index$_join$_001 | 800 | 15200 | 4 (0)| 00:00:01 |
10 |* 4 | HASH JOIN | | | | | |
11 | 5 | BITMAP CONVERSION TO ROWIDS| | 800 | 15200 | 1 (0)| 00:00:01 |
12 | 6 | BITMAP INDEX FULL SCAN | DESG_IDX | | | | |
13 | 7 | INDEX FAST FULL SCAN | STAFF_PK | 800 | 15200 | 4 (0)| 00:00:01 |
14 ----------
15
16 Predicate Information (identified by operation id):

7. DBA Security

The Company has captured the Social Security Numbers of the Staff to generate the payroll. Hence it is important to secure the Sensitive data such as SSN, to avoid the misuse of sensitive information.

We have used Data Redaction concept to mask the SSN Number for the users who accesses the Databases. This would help to maintain the data integrity.

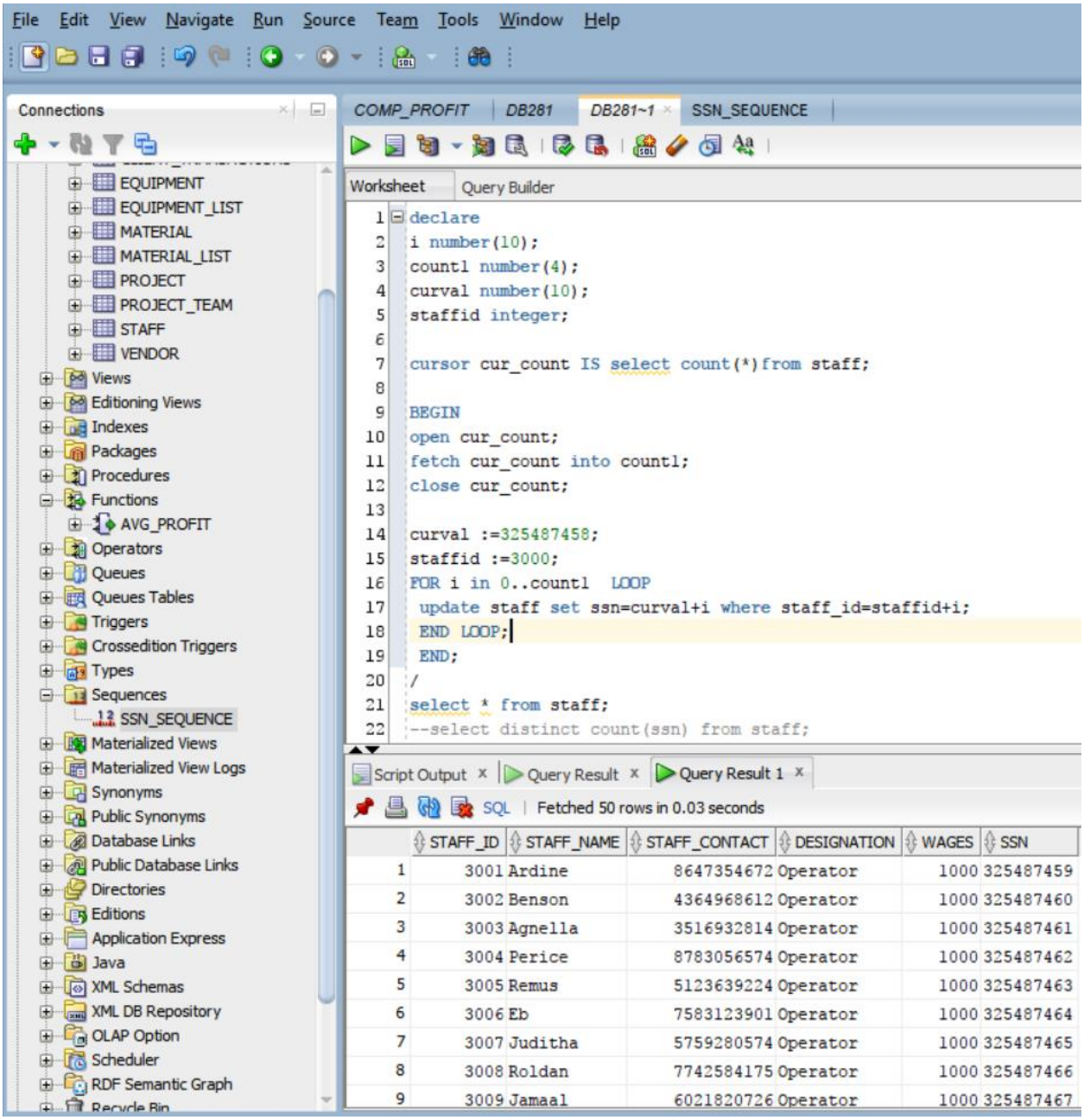
Step1: Creating a data redaction policy

```
BEGIN
 DBMS_REDACT.ADD_POLICY(
  object_schema => 'db281',
  object_name => 'staff',
  column_name => 'ssn',
  policy_name => 'db281_staff',
  function_type => DBMS_REDACT.FULL,
  function_parameters => '',
  expression => '1=1',
  policy_description => 'Redacts STAFF.',
  column_description => 'ssn contains Social Security Information');
END;
/
```

step2: Alter the redaction from FULL to PARTIAL

```
BEGIN
 DBMS_REDACT.ALTER_POLICY(
  object_schema => 'db281',
  object_name => 'staff',
  policy_name => 'db281_staff',
  action => DBMS_REDACT.MODIFY_COLUMN,
  column_name => 'ssn',
  function_type => DBMS_REDACT.PARTIAL,
  function_parameters => DBMS_REDACT.REDACT_US_SSN_F5);
END;
/
```

Step3: Without Redaction:



Step 4: with Data Redaction

```
35
36  SELECT * FROM redaction_values_for_type_full;
37  select * from staff;
```

Script Output | Query Result

SQL | Fetched 50 rows in 0.041 seconds

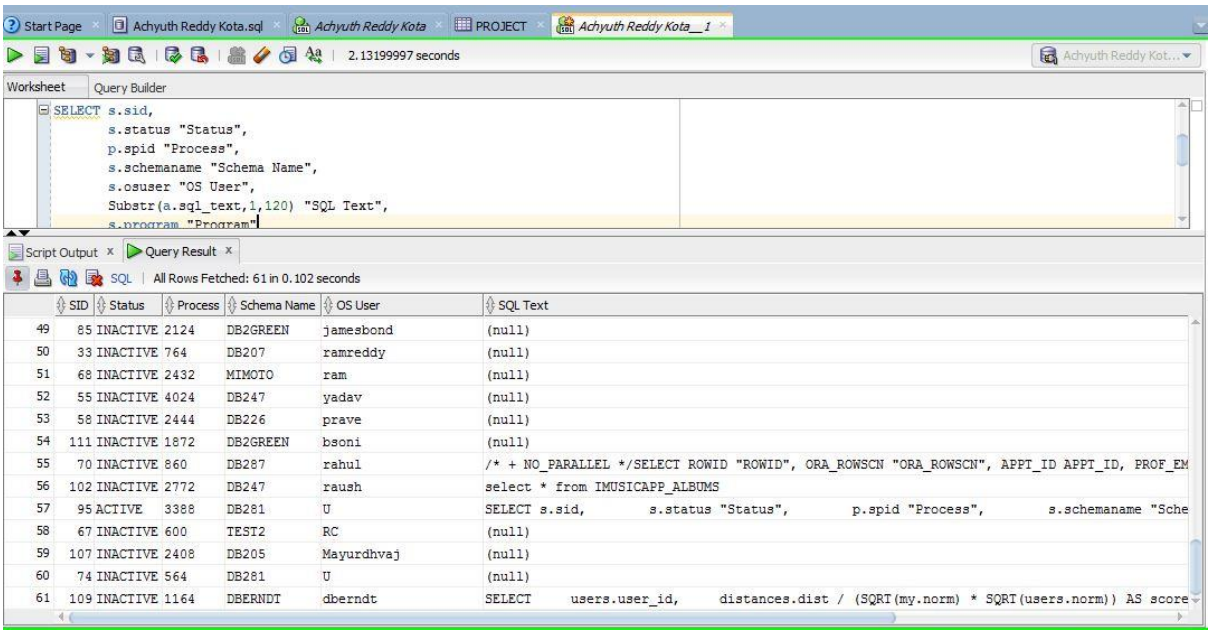
	STAFF_ID	STAFF_NAME	STAFF_CONTACT	DESIGNATION	WAGES	SSN
1	3001	Ardine	8647354672	Operator	1000	XXX-XX-59
2	3002	Benson	4364968612	Operator	1000	XXX-XX-60
3	3003	Agnella	3516932814	Operator	1000	XXX-XX-61
4	3004	Perice	8783056574	Operator	1000	XXX-XX-62
5	3005	Remus	5123639224	Operator	1000	XXX-XX-63
6	3006	Eb	7583123901	Operator	1000	XXX-XX-64
7	3007	Juditha	5759280574	Operator	1000	XXX-XX-65
8	3008	Roldan	7742584175	Operator	1000	XXX-XX-66
9	3009	Jamaal	6021820726	Operator	1000	XXX-XX-67
10	3010	Monah	6336548165	Operator	1000	XXX-XX-68
11	3011	Gordy	6308835486	Operator	1000	XXX-XX-69

8. DBA Scripts:

➢ Fetch Current Running Script:

Displays the SQL statements for currently running processes.

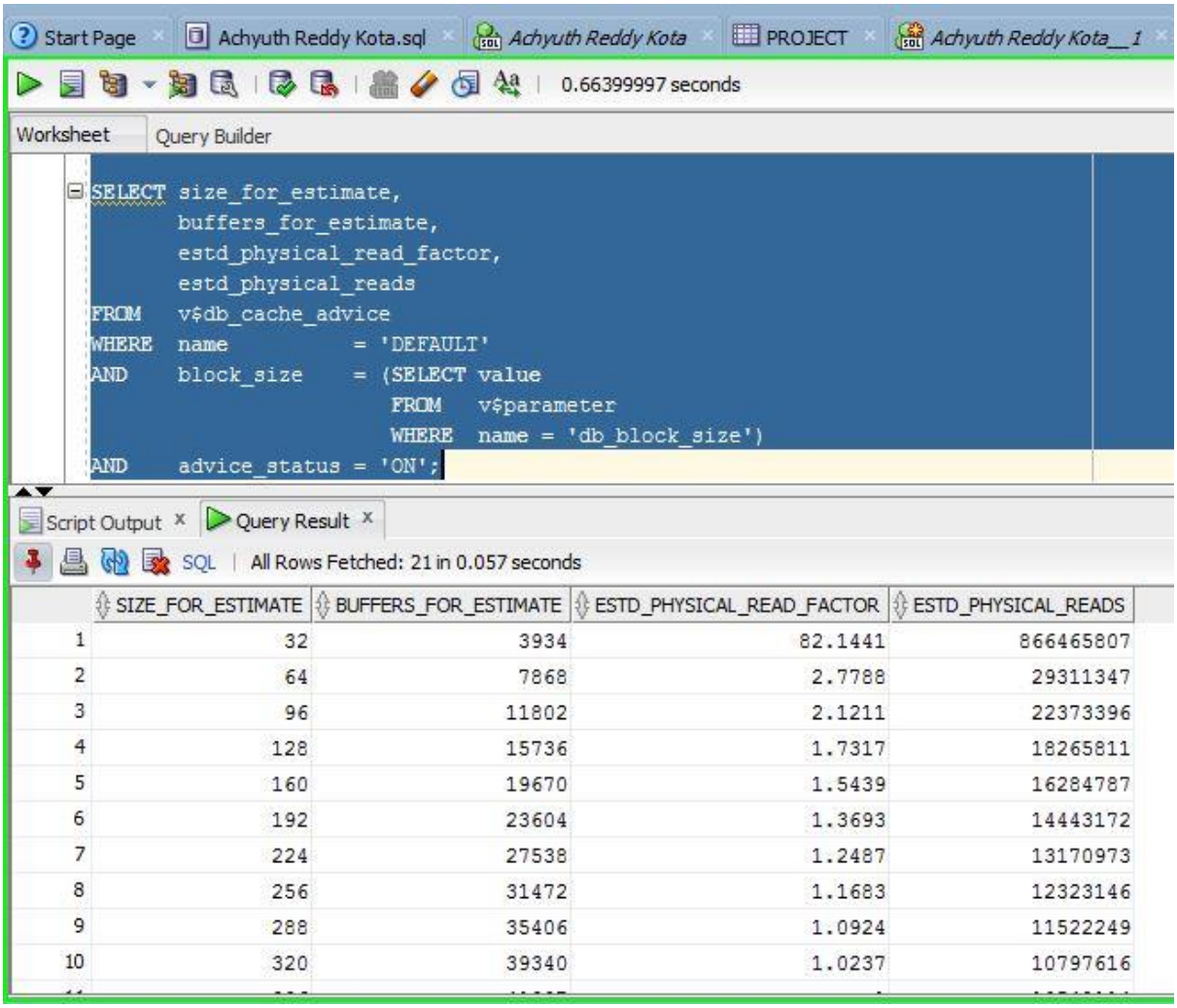
```
SET LINESIZE 500
SET PAGESIZE 1000
SET FEEDBACK OFF

SELECT s.sid,
       s.status "Status",
       p.spid "Process",
       s.schemaname "Schema Name",
       s.osuser "OS User",
       Substr(a.sql_text,1,120) "SQL Text",
       s.program "Program"
FROM   v$session s,
       v$sqlarea a,
       v$process p
WHERE  s.sql_hash_value = a.hash_value (+)
AND    s.sql_address    = a.address (+)
AND    s.paddr          = p.addr;

SET PAGESIZE 14
SET FEEDBACK ON
```



➢ Fetch Database Properties:

Display all database property values

```
1   COLUMN property_value FORMAT A50
2
3   SELECT property_name,
4          property_value
5   FROM   database_properties
6   ORDER BY property_name;
```



	SIZE_FOR_ESTIMATE	BUFFERS_FOR_ESTIMATE	ESTD_PHYSICAL_READ_FACTOR	ESTD_PHYSICAL_READS
1	32	3934	82.1441	866465807
2	64	7868	2.7788	29311347
3	96	11802	2.1211	22373396
4	128	15736	1.7317	18265811
5	160	19670	1.5439	16284787
6	192	23604	1.3693	14443172
7	224	27538	1.2487	13170973
8	256	31472	1.1683	12323146
9	288	35406	1.0924	11522249
10	320	39340	1.0237	10797616

➢ DBA Cache Advice:

This Script predicts how changes to the buffer cache will affect physical reads.

```
COLUMN size_for_estimate          FORMAT 999,999,999,999 heading 'Cache Size (MB)'
COLUMN buffers_for_estimate       FORMAT 999,999,999 heading 'Buffers'
COLUMN estd_physical_read_factor  FORMAT 999.90 heading 'Estd Phys|Read Factor'
COLUMN estd_physical_reads        FORMAT 999,999,999,999 heading 'Estd Phys| Reads'

SELECT size_for_estimate,
       buffers_for_estimate,
       estd_physical_read_factor,
       estd_physical_reads
FROM   v$db_cache_advice
WHERE  name          = 'DEFAULT'
AND    block_size    = (SELECT value
                        FROM   v$parameter
                        WHERE  name = 'db_block_size')
AND    advice_status = 'ON';
```

Start Page | Achyuth Reddy Kota.sql | Achyuth Reddy Kota | PROJECT | Achyuth Reddy Kota__1

0.66399997 seconds

Worksheet | Query Builder

```
SELECT size_for_estimate,
       buffers_for_estimate,
       estd_physical_read_factor,
       estd_physical_reads
FROM   v$db_cache_advice
WHERE  name          = 'DEFAULT'
AND    block_size    = (SELECT value
                        FROM   v$parameter
                        WHERE  name = 'db_block_size')
AND    advice_status = 'ON';
```

Script Output | Query Result

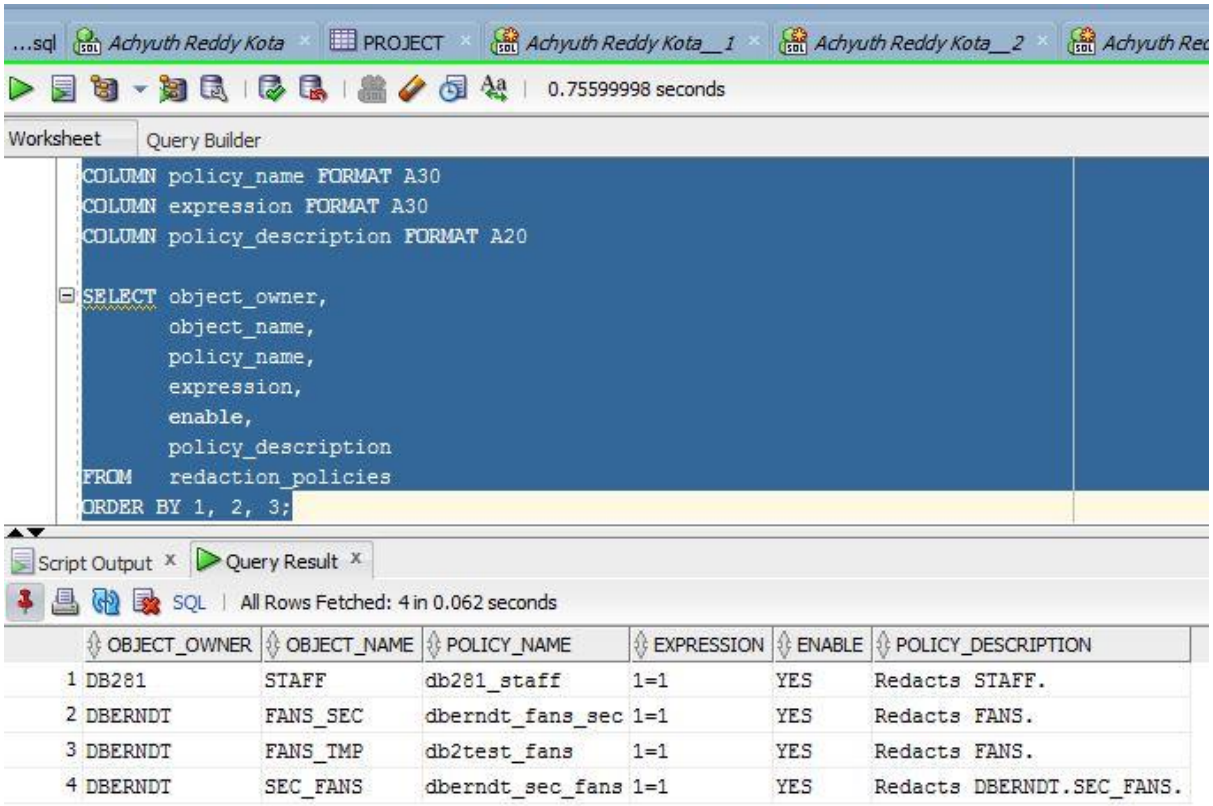
SQL | All Rows Fetched: 21 in 0.057 seconds

	SIZE_FOR_ESTIMATE	BUFFERS_FOR_ESTIMATE	ESTD_PHYSICAL_READ_FACTOR	ESTD_PHYSICAL_READS
1	32	3934	82.1441	866465807
2	64	7868	2.7788	29311347
3	96	11802	2.1211	22373396
4	128	15736	1.7317	18265811
5	160	19670	1.5439	16284787
6	192	23604	1.3693	14443172
7	224	27538	1.2487	13170973
8	256	31472	1.1683	12323146
9	288	35406	1.0924	11522249
10	320	39340	1.0237	10797616

➢ Fetch Data Redaction Policies:

This Script Displays redaction policy information

```
SET LINESIZE 200

COLUMN object_owner FORMAT A20
COLUMN object_name FORMAT A30
COLUMN policy_name FORMAT A30
COLUMN expression FORMAT A30
COLUMN policy_description FORMAT A20

SELECT object_owner,
       object_name,
       policy_name,
       expression,
       enable,
       policy_description
FROM   redaction_policies
ORDER BY 1, 2, 3;
```



...sql | Achyuth Reddy Kota | PROJECT | Achyuth Reddy Kota__1 | Achyuth Reddy Kota__2 | Achyuth Red

0.75599998 seconds

Worksheet | Query Builder

```
COLUMN policy_name FORMAT A30
COLUMN expression FORMAT A30
COLUMN policy_description FORMAT A20

SELECT object_owner,
       object_name,
       policy_name,
       expression,
       enable,
       policy_description
FROM   redaction_policies
ORDER BY 1, 2, 3;
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.062 seconds

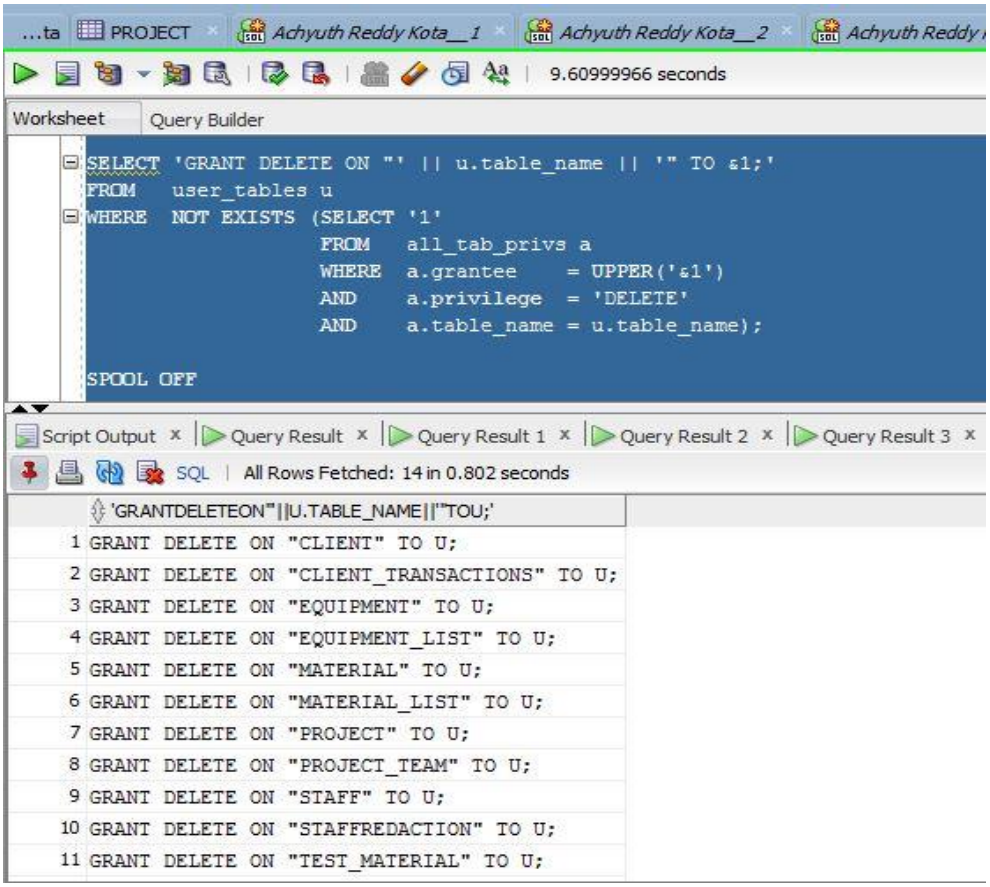
	OBJECT_OWNER	OBJECT_NAME	POLICY_NAME	EXPRESSION	ENABLE	POLICY_DESCRIPTION
1	DB281	STAFF	db281_staff	1=1	YES	Redacts STAFF.
2	DBERNDT	FANS_SEC	dberndt_fans_sec	1=1	YES	Redacts FANS.
3	DBERNDT	FANS_TMP	db2test_fans	1=1	YES	Redacts FANS.
4	DBERNDT	SEC_FANS	dberndt_sec_fans	1=1	YES	Redacts DBERNDT.SEC_FANS.

➢ Delete Grants:

This Script provide delete grants to a user U

```
SET PAGESIZE 0
SET FEEDBACK OFF
SET VERIFY OFF

SPOOL temp.sql

SELECT 'GRANT DELETE ON "' || u.table_name || '" TO &1;'
FROM   user_tables u
WHERE  NOT EXISTS (SELECT '1'
                   FROM   all_tab_privs a
                   WHERE  a.grantee    = UPPER('&1')
                   AND    a.privilege  = 'DELETE'
                   AND    a.table_name = u.table_name);

SPOOL OFF

@temp.sql

SET PAGESIZE 14
SET FEEDBACK ON
SET VERIFY ON
```

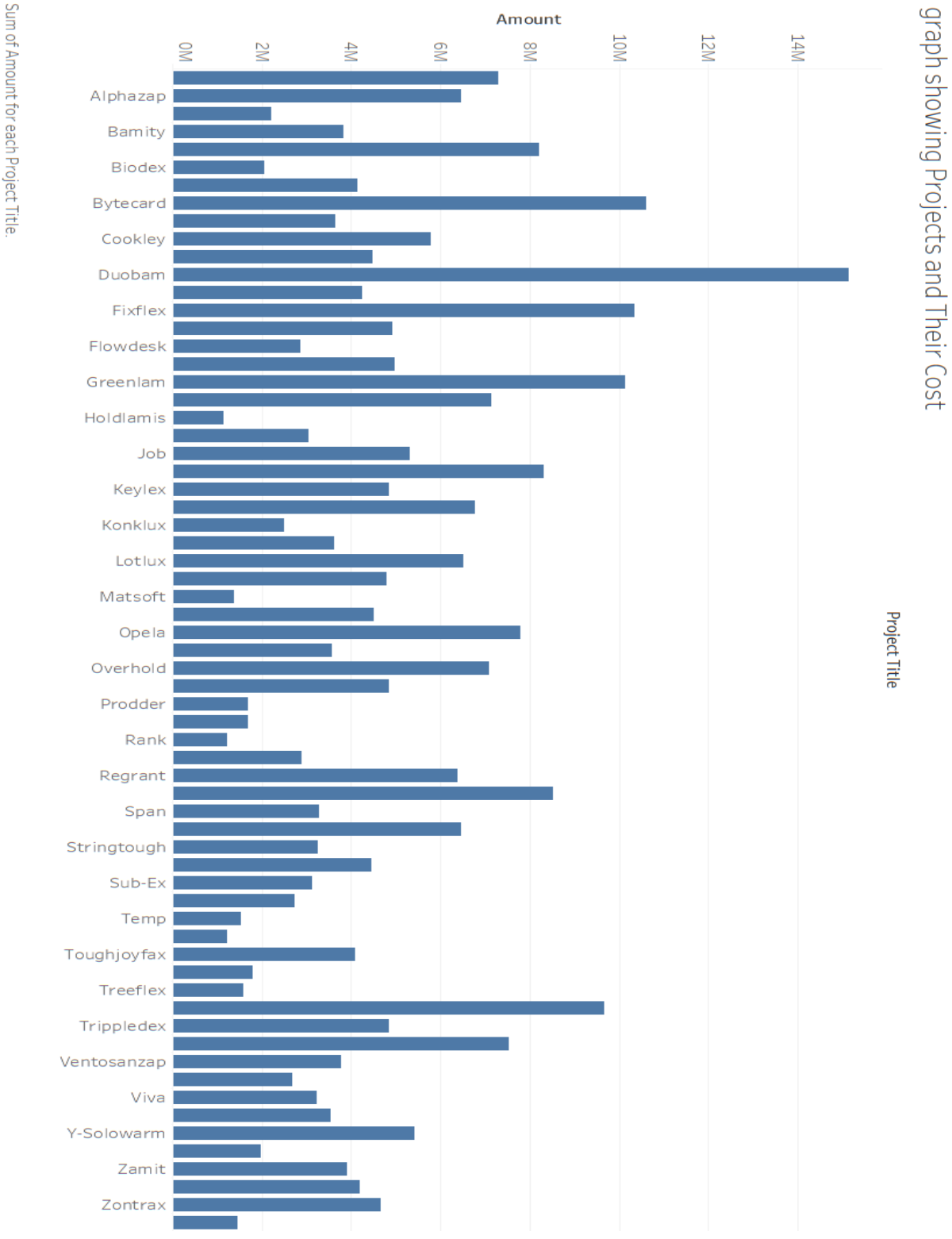
Z



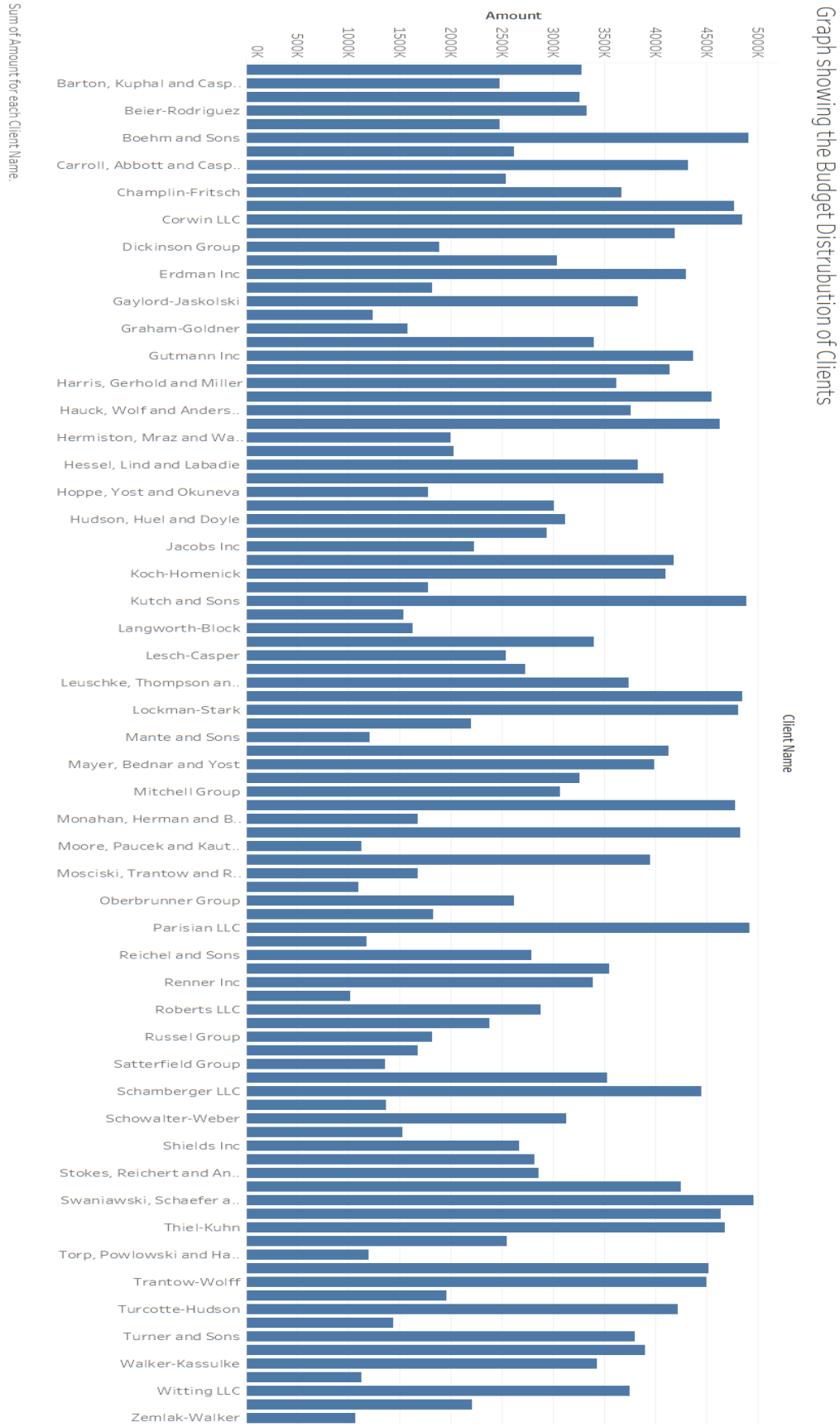
9. Data Visualization

Below are Some of the interesting Tableau visualizations of the Company's Data

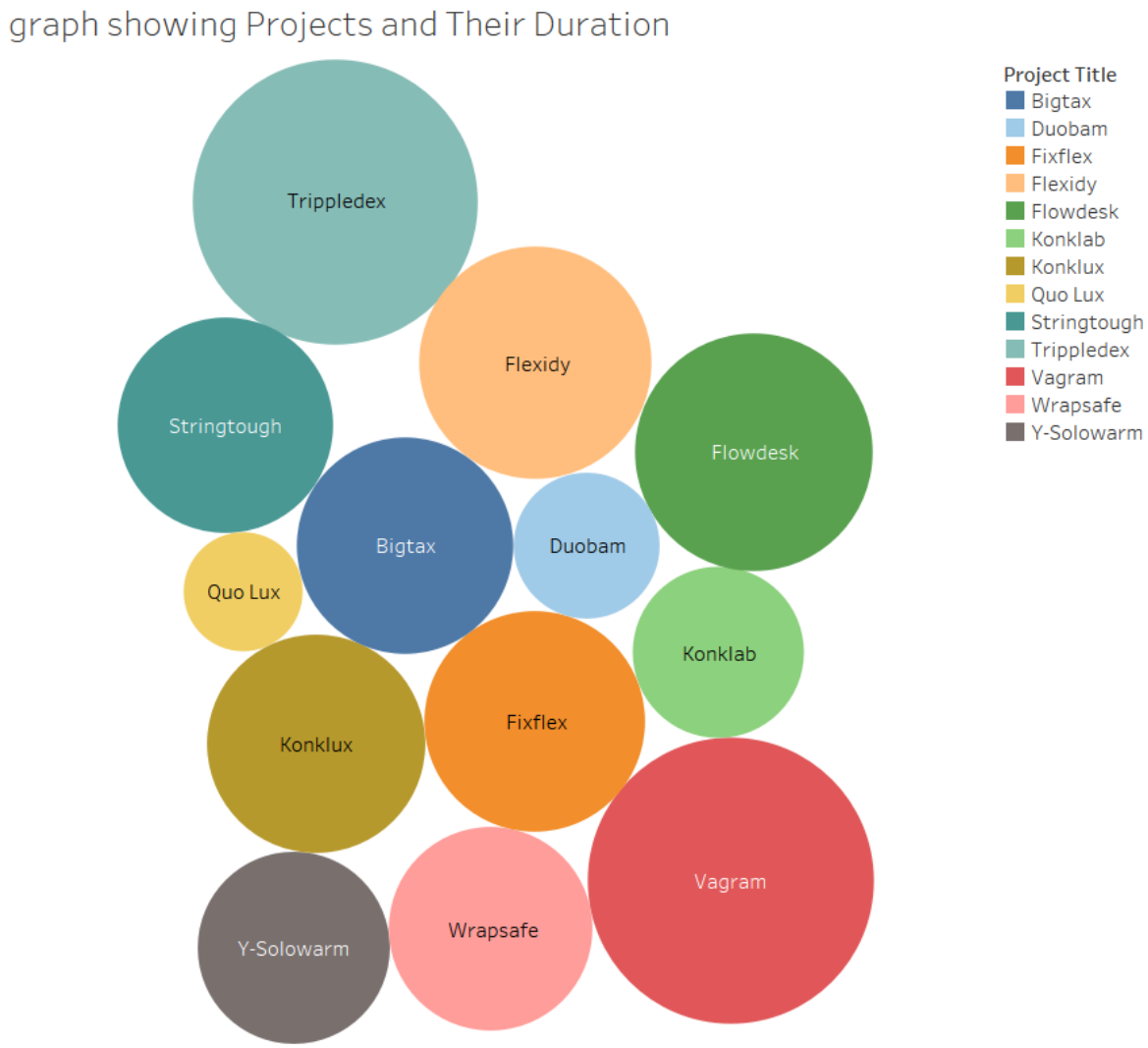
- **Budget Distribution of the Projects**



➢ Budget Distributions of the Clients



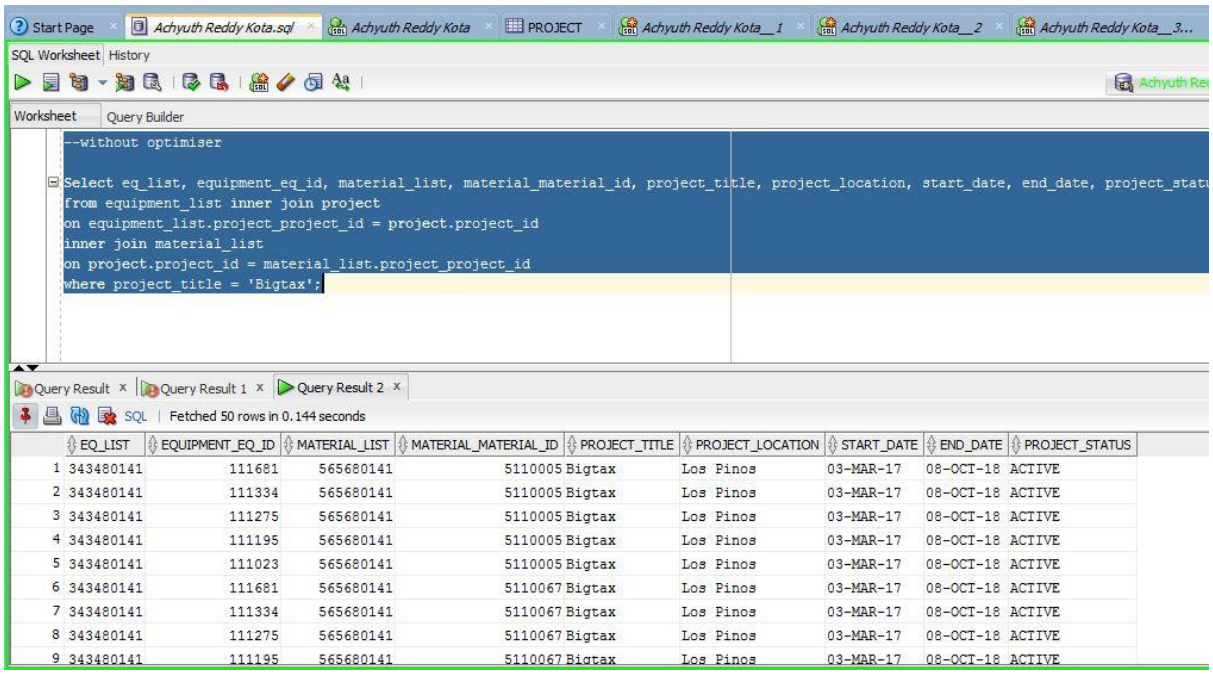
- **Visualiation of the Projects' Duration**
-



10. Optimizer - Performance Tuning

- **Query without Optimizer :**

```
Select eq_list, equipment_eq_id, material_list, material_material_id, project_title,
    project_location, start_date, end_date, project_status
from equipment_list inner join project
on equipment_list.project_project_id = project.project_id
inner join material_list
on project.project_id = material_list.project_project_id
where project_title = 'Bigtax';
```



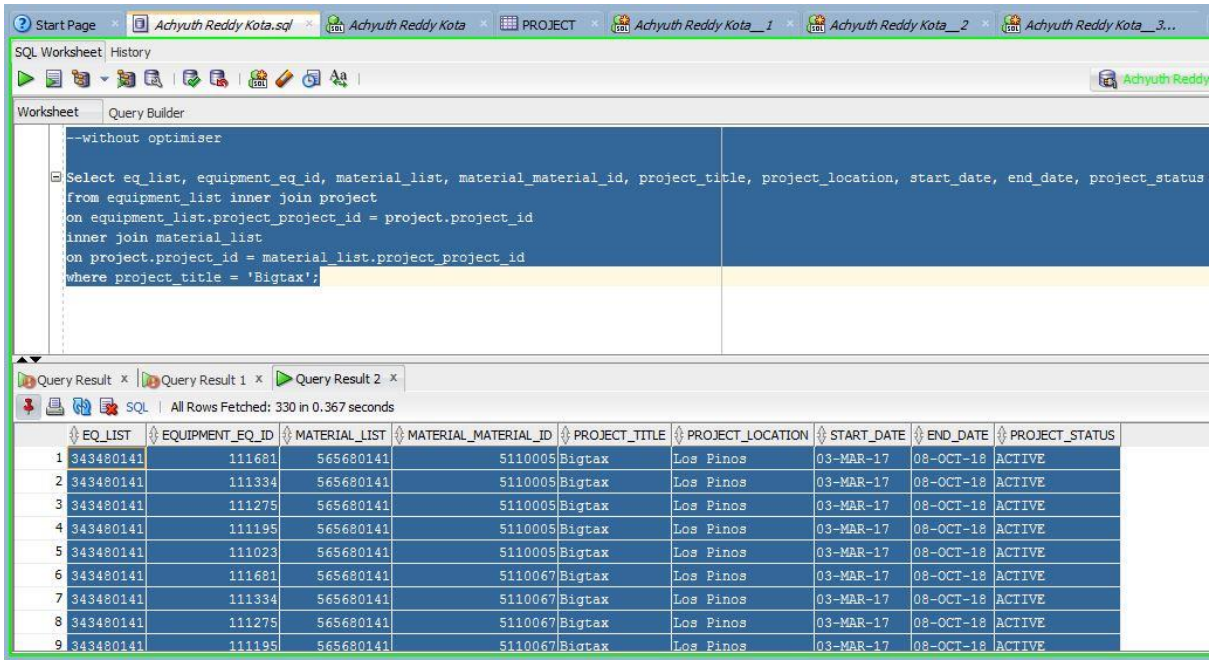
```
--without optimiser

Select eq_list, equipment_eq_id, material_list, material_material_id, project_title, project_location, start_date, end_date, project_statu
from equipment_list inner join project
on equipment_list.project_project_id = project.project_id
inner join material_list
on project.project_id = material_list.project_project_id
where project_title = 'Bigtax';
```

Query Result 2 — Fetched 50 rows in 0.144 seconds

	EQ_LIST	EQUIPMENT_EQ_ID	MATERIAL_LIST	MATERIAL_MATERIAL_ID	PROJECT_TITLE	PROJECT_LOCATION	START_DATE	END_DATE	PROJECT_STATUS
1	343480141	111681	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
2	343480141	111334	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
3	343480141	111275	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
4	343480141	111195	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
5	343480141	111023	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
6	343480141	111681	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
7	343480141	111334	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
8	343480141	111275	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
9	343480141	111195	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE

50 Rows fetched in 0.144 seconds

Without Optimizer all 330rows fetched in 0.367 seconds



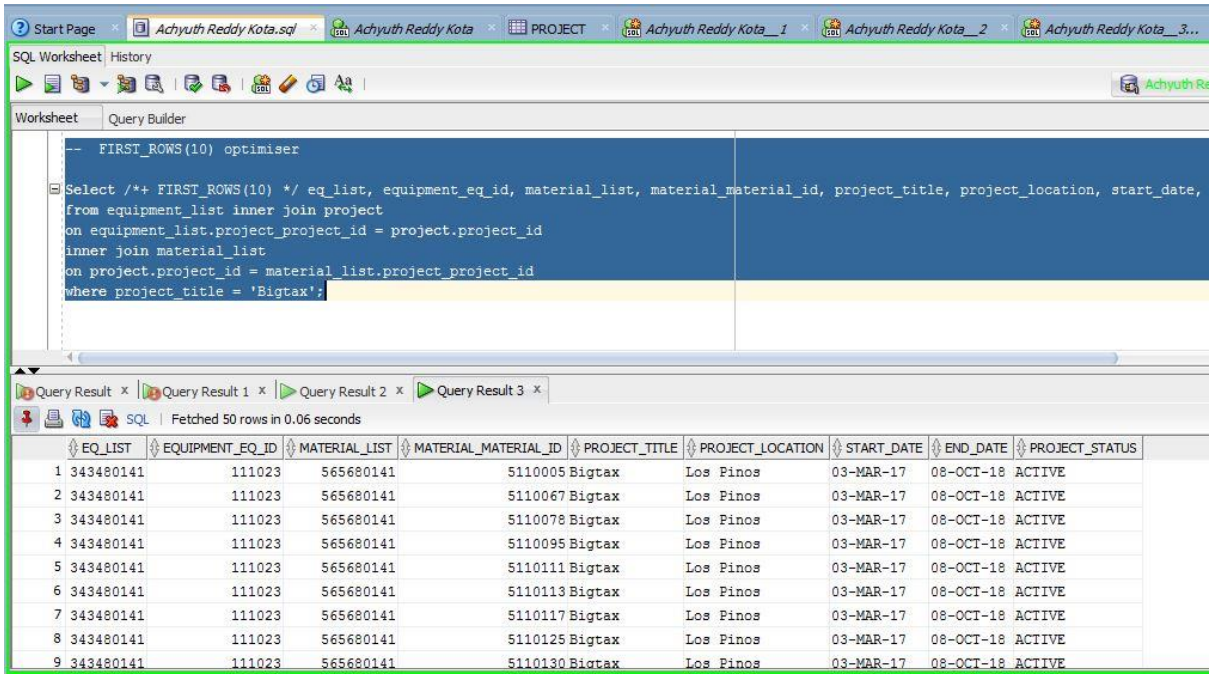
```
--without optimiser

Select eq_list, equipment_eq_id, material_list, material_material_id, project_title, project_location, start_date, end_date, project_status
from equipment_list inner join project
on equipment_list.project_project_id = project.project_id
inner join material_list
on project.project_id = material_list.project_project_id
where project_title = 'Bigtax';
```

All Rows Fetched: 330 in 0.367 seconds

	EQ_LIST	EQUIPMENT_EQ_ID	MATERIAL_LIST	MATERIAL_MATERIAL_ID	PROJECT_TITLE	PROJECT_LOCATION	START_DATE	END_DATE	PROJECT_STATUS
1	343480141	111681	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
2	343480141	111334	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
3	343480141	111275	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
4	343480141	111195	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
5	343480141	111023	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
6	343480141	111681	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
7	343480141	111334	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
8	343480141	111275	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
9	343480141	111195	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE

➢ Query Optimization with first N Rows

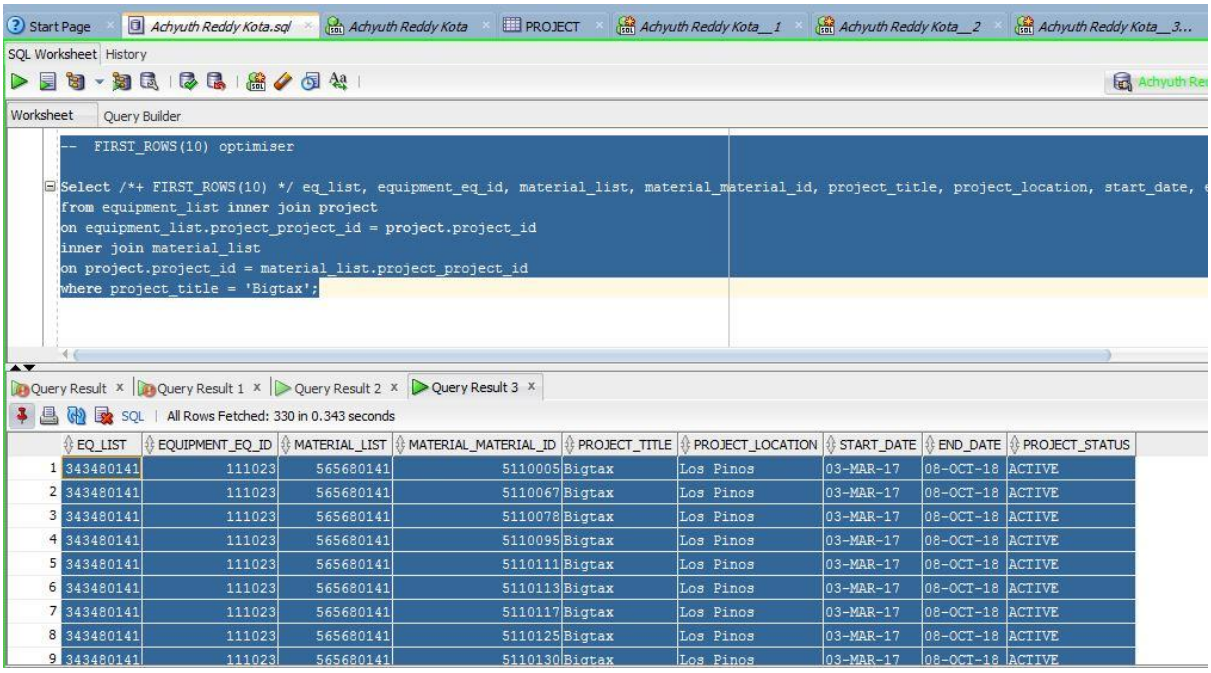
```
FIRST_ROWS(10) optimiser

Select /*+ FIRST_ROWS(10) */ eq_list, equipment_eq_id, material_list,
material_material_id, project_title, project_location, start_date, end_date, project_status
from equipment_list inner join project
on equipment_list.project_project_id = project.project_id
inner join material_list
on project.project_id = material_list.project_project_id
where project_title = 'Bigtax';
```



	EQ_LIST	EQUIPMENT_EQ_ID	MATERIAL_LIST	MATERIAL_MATERIAL_ID	PROJECT_TITLE	PROJECT_LOCATION	START_DATE	END_DATE	PROJECT_STATUS
1	343480141	111023	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
2	343480141	111023	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
3	343480141	111023	565680141	5110078	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
4	343480141	111023	565680141	5110095	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
5	343480141	111023	565680141	5110111	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
6	343480141	111023	565680141	5110113	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
7	343480141	111023	565680141	5110117	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
8	343480141	111023	565680141	5110125	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
9	343480141	111023	565680141	5110130	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE

50 Rows fetched in 0.06 seconds using First 10 rows optimizer

Using First N row optimizer all 330 rows were fetched in 0.343 seconds



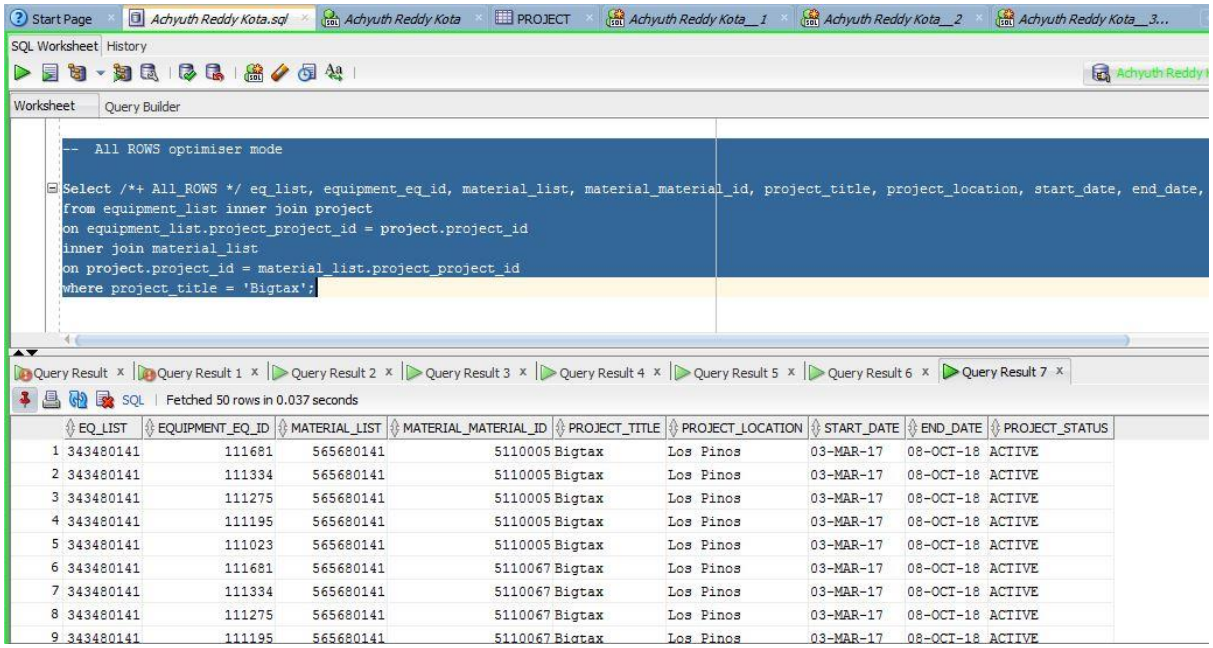
```
--  FIRST_ROWS(10) optimiser

Select /*+ FIRST_ROWS(10) */ eq_list, equipment_eq_id, material_list, material_material_id, project_title, project_location, start_date,
from equipment_list inner join project
on equipment_list.project_project_id = project.project_id
inner join material_list
on project.project_id = material_list.project_project_id
where project_title = 'Bigtax';
```

All Rows Fetched: 330 in 0.343 seconds

	EQ_LIST	EQUIPMENT_EQ_ID	MATERIAL_LIST	MATERIAL_MATERIAL_ID	PROJECT_TITLE	PROJECT_LOCATION	START_DATE	END_DATE	PROJECT_STATUS
1	343480141	111023	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
2	343480141	111023	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
3	343480141	111023	565680141	5110078	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
4	343480141	111023	565680141	5110095	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
5	343480141	111023	565680141	5110111	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
6	343480141	111023	565680141	5110113	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
7	343480141	111023	565680141	5110117	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
8	343480141	111023	565680141	5110125	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
9	343480141	111023	565680141	5110130	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE

➢ Optimizing All Rows

```
1  Select /*+ All_ROWS */ eq_list, equipment_eq_id, material_list, material_material_id,
2  project_title, project_location, start_date, end_date, project_status
3  from equipment_list inner join project
4  on equipment_list.project_project_id = project.project_id
5  inner join material_list
6  on project.project_id = material_list.project_project_id
7  where project_title = 'Bigtax';
```

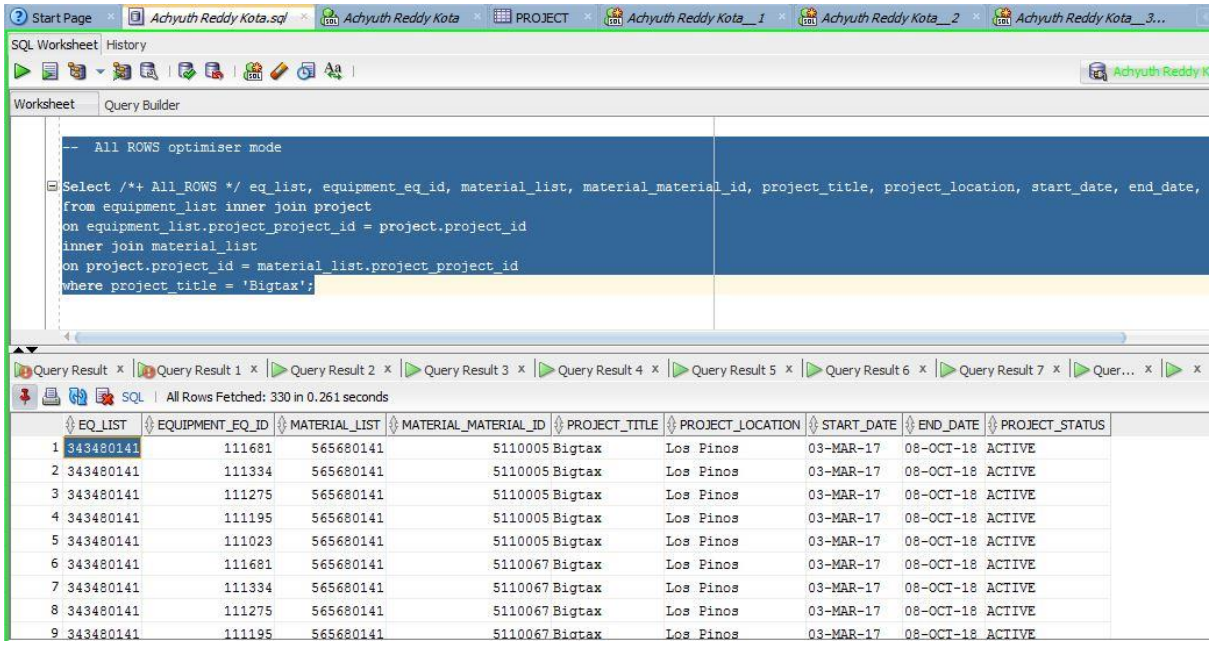


```
-- All ROWS optimiser mode

Select /*+ All_ROWS */ eq_list, equipment_eq_id, material_list, material_material_id, project_title, project_location, start_date, end_date,
from equipment_list inner join project
on equipment_list.project_project_id = project.project_id
inner join material_list
on project.project_id = material_list.project_project_id
where project_title = 'Bigtax';
```

Fetched 50 rows in 0.037 seconds

	EQ_LIST	EQUIPMENT_EQ_ID	MATERIAL_LIST	MATERIAL_MATERIAL_ID	PROJECT_TITLE	PROJECT_LOCATION	START_DATE	END_DATE	PROJECT_STATUS
1	343480141	111681	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
2	343480141	111334	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
3	343480141	111275	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
4	343480141	111195	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
5	343480141	111023	565680141	5110005	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
6	343480141	111681	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
7	343480141	111334	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
8	343480141	111275	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
9	343480141	111195	565680141	5110067	Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE

Using All row optimizer 50rows were fetched in 0.037 seconds



All Rows were fetched in 0.261 seconds