

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	3
2.	Database Design.....	4
3.	Physical Database Design.....	6
4.	Query Writing.....	20
5.	Stored Procedure Programming	23
6.	Performance Tuning.....	26
7.	DBA Security.....	28
8.	DBA Scripts:.....	31
9.	Data Visualization	36
10.	Optimizer - Performance Tuning.....	39

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:

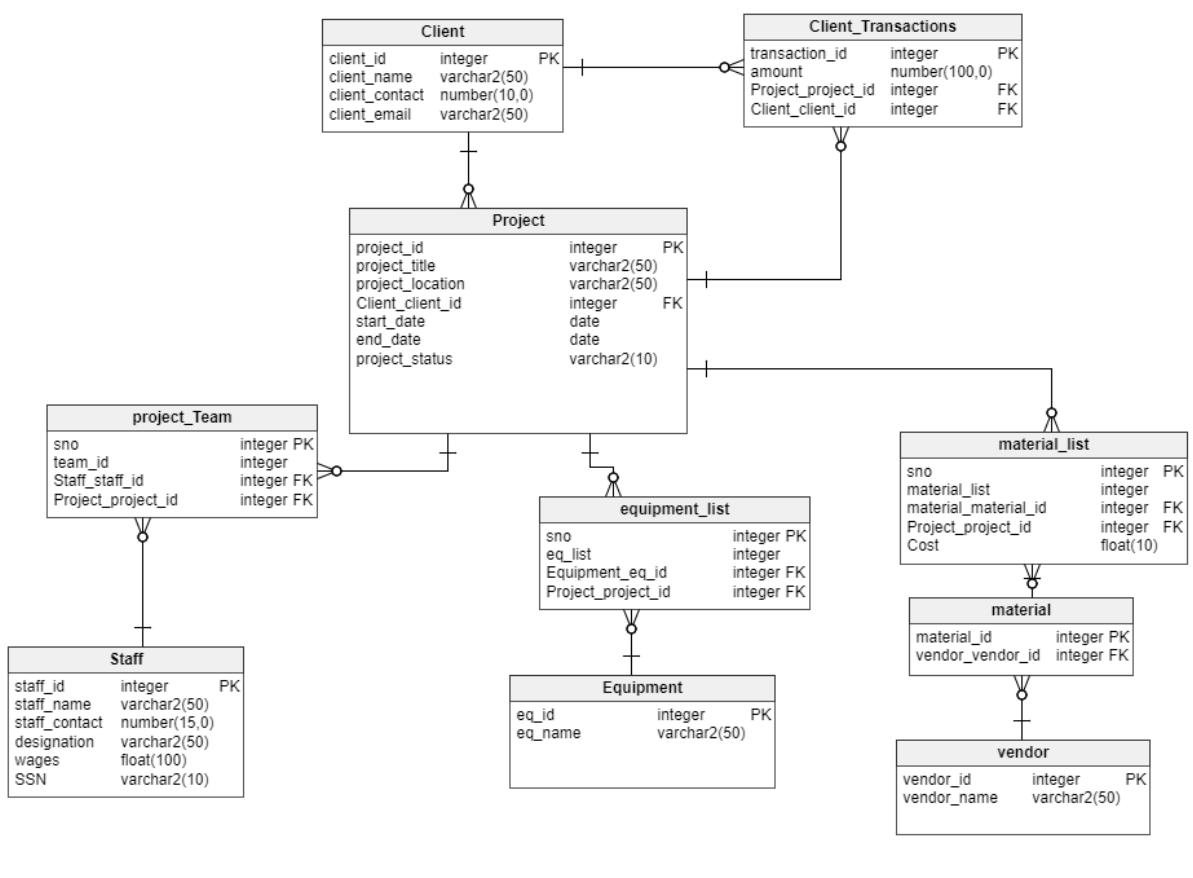
1. Each client can offer many projects to the company.
2. Client pays for the project cost in one installment.
3. Wages of the staff are recorded per project period.
4. Each project has a list of materials and equipment being used.

2. Database Design

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

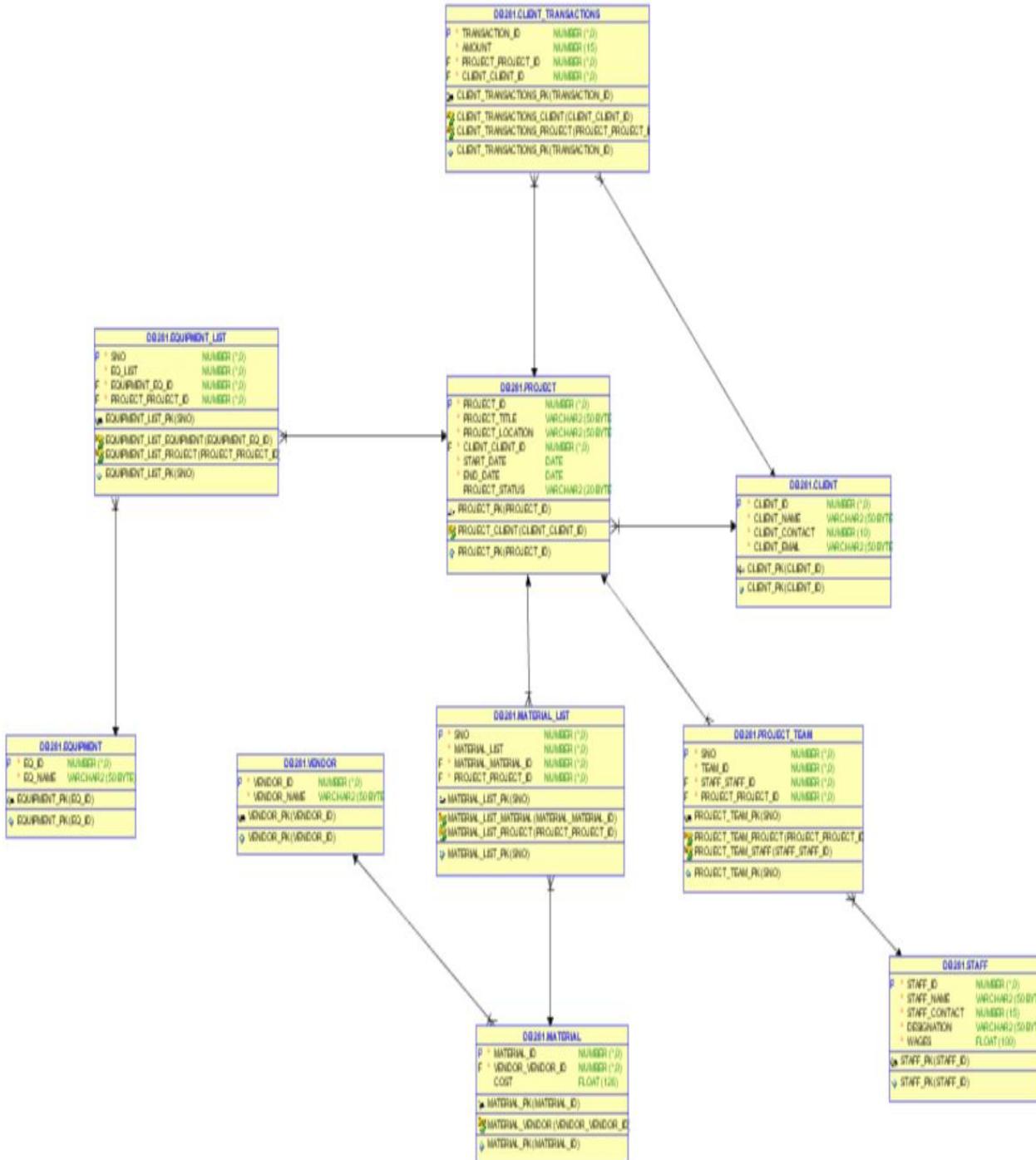
Conceptual Design:

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)	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						
	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

Constraints										
	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);
```

Constraints									
	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

Dependencies						
	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

Indexes										
	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)
);
```

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

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

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

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						
	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)
) ;
```

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

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

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	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)
) ;
```

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

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

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)
) ;
```

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);
```

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

Actions...										
INDEX_OWNER	INDEX_NAME	UNIQUENESS	STATUS	INDEX_TYPE	TEMPORARY	PARTITIONED	FUNCIDX_STATUS	JOIN_INDEX	COLUMNS	
1 DB201	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)
) ;
```

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 (project_id);
```

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

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

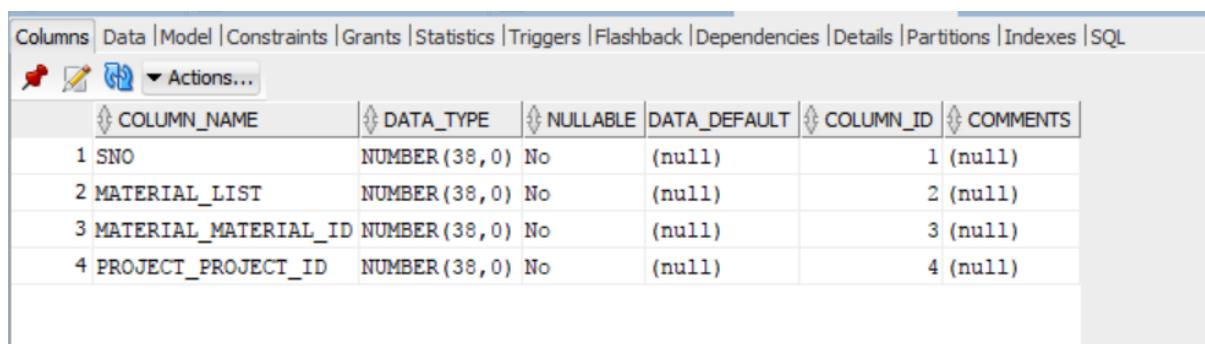
Description: This contains the List of materials used for a product

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)
);
```



The screenshot shows the 'Columns' tab of the Oracle Database SQL Developer interface. It displays the structure of the 'material_list' table with four columns: SNO, MATERIAL_LIST, MATERIAL_MATERIAL_ID, and PROJECT_PROJECT_ID. Each column is defined as a NUMBER(38,0) type and is marked as NOT NULL. The primary key constraint is defined on the SNO column.

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);
```

Constraints							
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

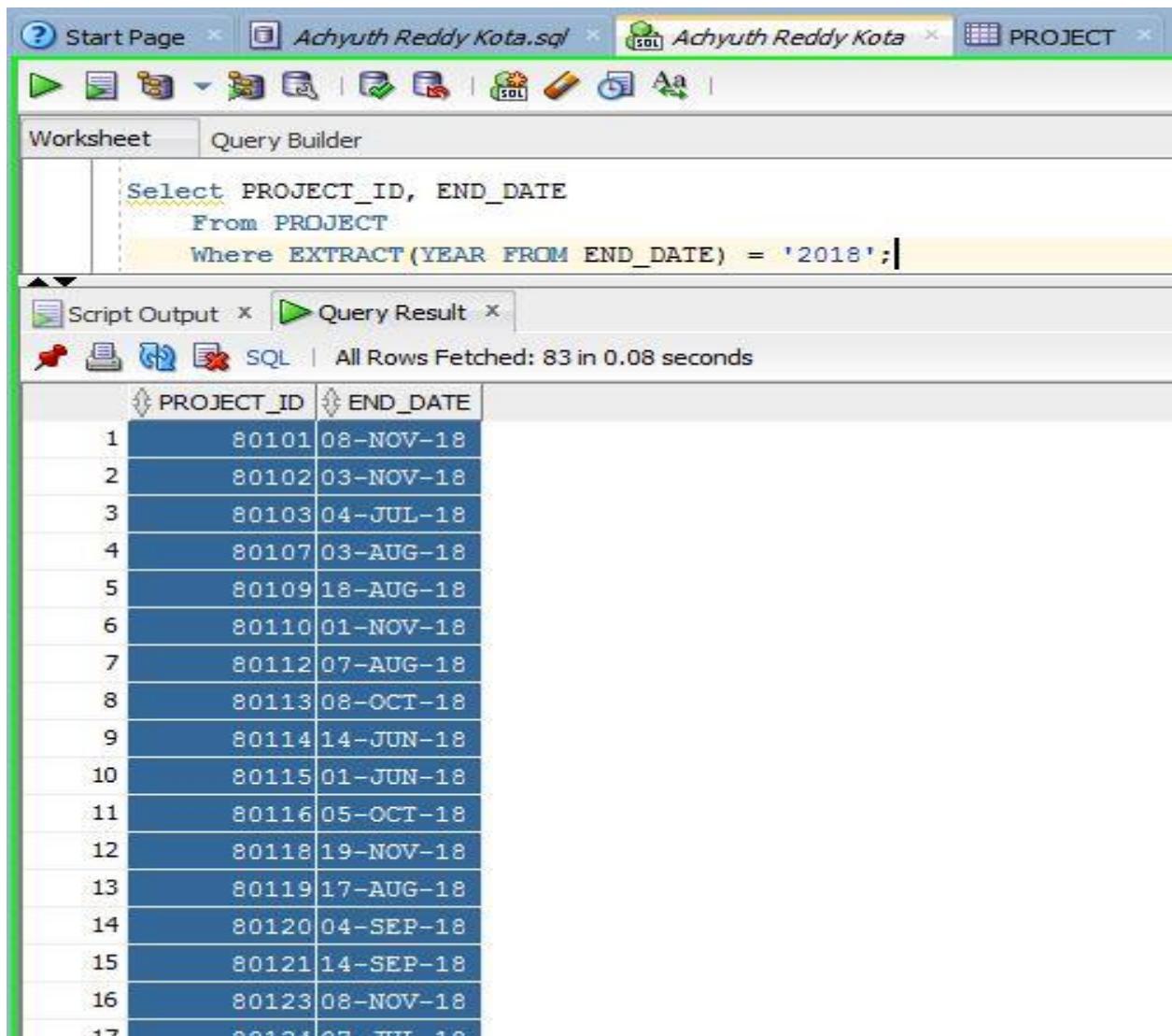
3.10.4 Indexes

Indexes										
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 screenshot shows the Oracle SQL Developer interface. The top menu bar includes 'Start Page', 'Achyuth Reddy Kota.sql', 'Achyuth Reddy Kota', and 'PROJECT'. Below the menu is a toolbar with various icons. The main area has tabs for 'Worksheet' and 'Query Builder', with 'Worksheet' selected. A SQL query is written in the worksheet:

```
Select PROJECT_ID, END_DATE  
From PROJECT  
Where EXTRACT(YEAR FROM END_DATE) = '2018';
```

Below the query, the 'Query Result' tab is active, showing the results of the executed query. The results are presented in a grid:

PROJECT_ID	END_DATE
1	08-NOV-18
2	03-NOV-18
3	04-JUL-18
4	03-AUG-18
5	18-AUG-18
6	01-NOV-18
7	07-AUG-18
8	08-OCT-18
9	14-JUN-18
10	01-JUN-18
11	05-OCT-18
12	19-NOV-18
13	17-AUG-18
14	04-SEP-18
15	14-SEP-18
16	08-NOV-18
17	07-JUL-18

Below the grid, a status message indicates 'All Rows Fetched: 83 in 0.08 seconds'.

- 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:

The screenshot shows a MySQL Workbench interface. In the top-left pane, there is a SQL editor window containing the following code:

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

In the bottom-right pane, there is a "Query Result" table with the following data:

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.

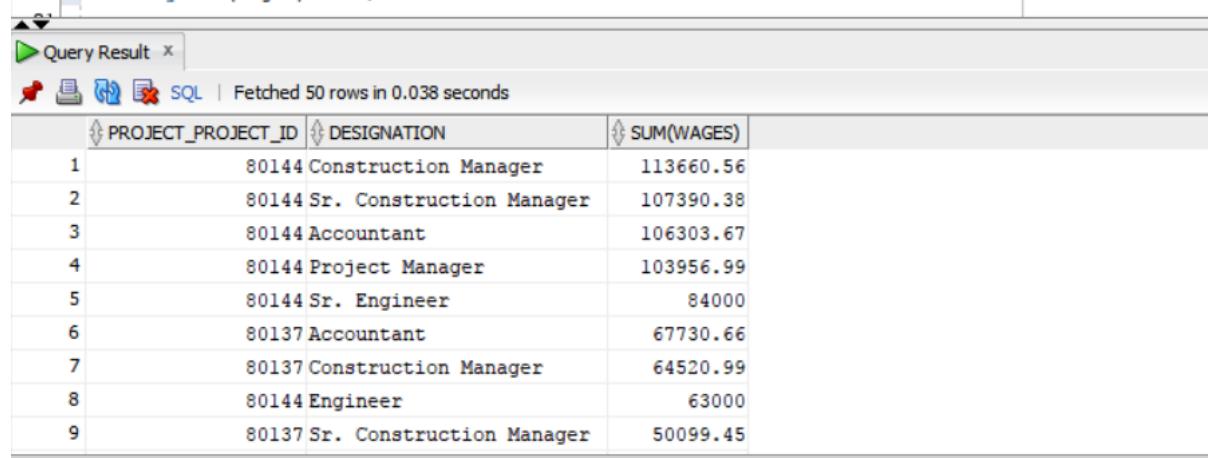
```

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

```

Results:

16
 17 select pt.PROJECT_PROJECT_ID, designation,sum(wages) from staff s inner join project_team pt
 18 on s.staff_id=pt.STAFF_STAFF_ID
 19 group by pt.PROJECT_PROJECT_ID,s.DESIGNATION
 20 order by sum(wages) desc;



	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.

```
1  create or replace PROCEDURE COMP_PROFIT
2    (in_project_id      project.project_id%TYPE)
3  AS
4
5    CURSOR CUR_WAGES(in_project_id project.project_id%TYPE) IS
6      SELECT sum( distinct st.wages) from staff st, project_team pt
7      WHERE st.STAFF_ID=pt.STAFF_STAFF_ID
8      AND pt.PROJECT_PROJECT_ID=in_project_id;
9
10   CURSOR CUR_MAT_COST(in_project_id project.project_id%TYPE) IS
11     SELECT sum(distinct m.COST) from MATERIAL_LIST ml ,project p,MATERIAL m
12     WHERE ml.PROJECT_PROJECT_ID=p.PROJECT_ID
13     AND ml.MATERIAL_MATERIAL_ID=m.MATERIAL_ID
14     AND p.project_id=in_project_id;
15
16   CURSOR CUR_AMOUNT IS
17     SELECT ct.AMOUNT from CLIENT_TRANSACTIONS ct where project_PROJECT_ID=in_project_id;
18
19   out_mat_cost FLOAT;
20   out_wage FLOAT;
21   out_amount FLOAT;
22   out_profit FLOAT;
23
24   BEGIN
25
26
27   OPEN CUR_WAGES(in_project_id);
28   FETCH CUR_WAGES into out_wage;
29   CLOSE CUR_WAGES;
30
31   OPEN CUR_MAT_COST(in_project_id);
32   FETCH CUR_MAT_COST into out_mat_cost;
33   CLOSE CUR_MAT_COST;
34
35   open cur_amount;
36   FETCH cur_amount into out_amount;
37   CLOSE cur_amount;
38
39   out_profit :=(out_amount)-(out_mat_cost+out_wage);
40
41   EXCEPTION
42   WHEN OTHERS THEN
43     dbms_output.put_line ('Error while calculsting the ');
44
45   DBMS_OUTPUT.PUT_LINE ('The human cost of the project is'|| out_wage);
46   DBMS_OUTPUT.PUT_LINE ('The material cost of the project is'|| out_mat_cost);
47   DBMS_OUTPUT.PUT_LINE ('The total budget of the project is'|| out_amount);
48   DBMS_OUTPUT.PUT_LINE ('The total profit of the company is'||out_profit);
49
50 END;
```

Results:

The screenshot shows the Oracle SQL Developer interface. On the left is a code editor window titled 'COMP_PROFIT' containing PL/SQL code. On the right is a 'Run PL/SQL' dialog box. The dialog has a 'Target' section set to 'COMP_PROFIT' and a 'Parameters' section with one entry: 'IN_PROJECT_ID' of type NUMBER with value 80125. The 'PL/SQL Block' pane contains the following code:

```
16 CURSOR CUR_AMOUNT IS
17   SELECT ct.AMOUNT from (
18     out_mat_cost FLOAT;
19     out_wage FLOAT;
20     out_amount FLOAT;
21     out_profit FLOAT;
22
23   BEGIN
24
25   OPEN CUR_WAGES(in_project_id);
26   FETCH CUR_WAGES into out_wage;
27   CLOSE CUR_WAGES;
28
29   OPEN CUR_MAT_COST(in_project_id);
30   FETCH CUR_MAT_COST into out_mat_cost;
31   CLOSE CUR_MAT_COST;
32
33   open cur_amount;
34   FETCH cur_amount into out_amount;
35   CLOSE cur_amount;
36
37   out_profit :=(out_amount - out_mat_cost) * out_wage;
38
39   EXCEPTION
40   WHEN OTHERS THEN
41     dbms_output.put_line('An error occurred');
42
43   DBMS_OUTPUT.PUT_LINE ('The human cost of the project is'|| out_wage);
44   DBMS_OUTPUT.PUT_LINE ('The material cost of the project is'|| out_mat_cost);
45   DBMS_OUTPUT.PUT_LINE ('The total budget of the project is'|| out_amount);
46   DBMS_OUTPUT.PUT_LINE ('The total profit of the company is'||out_profit);
47
48 END;
```

The 'Run PL/SQL' dialog also includes buttons for 'Save File...', 'From File...', 'Reset', 'OK', and 'Cancel'.

The screenshot shows the 'Log' window of Oracle SQL Developer. It displays the output of the executed PL/SQL block. The log message is as follows:

```
Running: McConnections%23DB281.jpr - Log
Connecting to the database DB281.
The human cost of the project is155647.23
The material cost of the project is1167480.95
The total budget of the project is2843921
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

```
1  create or replace FUNCTION AVG_PROFIT
2
3  RETURN float IS
4      avg_profit float := 0;
5
6  out_avg_wage    FLOAT;
7  out_avg_mat     FLOAT;
8  out_avg_amount  FLOAT;
9
10 cursor ave_wage IS select avg(wages) from STAFF;
11 cursor avg_cost  is select avg(cost)   from MATERIAL;
12 cursor avg_amount is select avg(amount) from CLIENT_TRANSACTIONS;
13
14 BEGIN
15 open ave_wage;
16 fetch ave_wage into out_avg_wage;
17 close ave_wage;
18
19 open avg_cost;
20 fetch avg_cost into out_avg_mat;
21 close avg_cost;
22
23 open avg_amount;
24 fetch avg_amount into out_avg_amount;
25 avg_profit :=out_avg_amount-(out_avg_wage+out_avg_amount);
26 END;
```

6. Performance Tuning

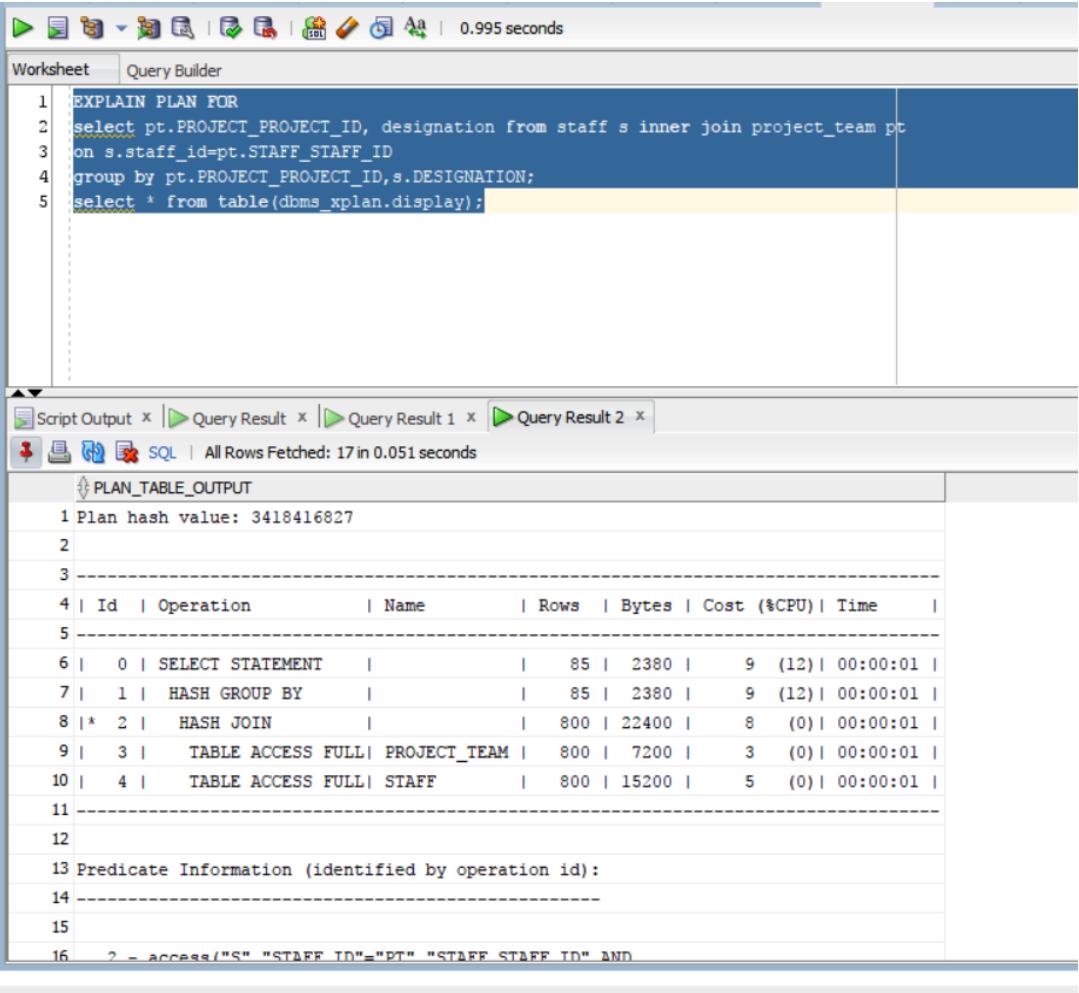
➤ Indexing:

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:



The screenshot shows the Oracle SQL Developer interface. The top window is a Worksheet with the title 'Query Builder'. It contains the EXPLAIN PLAN SQL code. Below it is a 'PLAN_TABLE_OUTPUT' window displaying the execution plan. The plan shows a HASH JOIN operation (id 8) that uses a TABLE ACCESS FULL on the 'PROJECT_TEAM' table (id 3) and a TABLE ACCESS FULL on the 'STAFF' table (id 4). The total cost is 8 (0%). The bottom status bar indicates 'All Rows Fetched: 17 in 0.051 seconds'.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		85	2380	9 (12)	00:00:01
1	HASH GROUP BY		85	2380	9 (12)	00:00:01
2	HASH JOIN		800	22400	8 (0)	00:00:01
3	TABLE ACCESS FULL	PROJECT_TEAM	800	7200	3 (0)	00:00:01
4	TABLE ACCESS FULL	STAFF	800	15200	5 (0)	00:00:01

With Tuning:

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, the connection is set to 'DB281~10' and the schema is 'STAFF'. Below the toolbar, there are tabs for 'Worksheet' and 'Query Builder', with 'Worksheet' selected. The main workspace contains the following SQL code:

```
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);
```

Below the code, the status bar indicates 'All Rows Fetched: 21 in 0.057 seconds'. The bottom half of the window displays the 'PLAN_TABLE_OUTPUT' section, which provides a detailed breakdown of the execution plan:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		800	22400	7 (0)	00:00:01
1	HASH JOIN		800	22400	7 (0)	00:00:01
2	TABLE ACCESS FULL	PROJECT_TEAM	800	7200	3 (0)	00:00:01
3	VIEW	index\$_join\$_001	800	15200	4 (0)	00:00:01
4	HASH JOIN					
5	BITMAP CONVERSION TO ROWIDS		800	15200	1 (0)	00:00:01
6	BITMAP INDEX FULL SCAN	DESG_IDX				
7	INDEX FAST FULL SCAN	STAFF_PK	800	15200	4 (0)	00:00:01

At the bottom of the output, it says '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

```
1 BEGIN
2   DBMS_REDACT.ADD_POLICY(
3     object_schema => 'db281',
4     object_name => 'staff',
5     column_name => 'ssn',
6     policy_name => 'db281_staff',
7     function_type => DBMS_REDACT.FULL,
8     function_parameters => '',
9     expression => '1=1',
10    policy_description => 'Redacts STAFF.',
11    column_description => 'ssn contains Social Security Information');
12  END;
13 |
```

step2: Alter the redaction from FULL to PARTIAL

```
1 BEGIN
2   DBMS_REDACT.ALTER_POLICY(
3     object_schema => 'db281',
4     object_name => 'staff',
5     policy_name => 'db281_staff',
6     action => DBMS_REDACT.MODIFY_COLUMN,
7     column_name => 'ssn',
8     function_type => DBMS_REDACT.PARTIAL,
9     function_parameters => DBMS_REDACT.REDACT_US_SSN_F5);
10  END;
11 |
```

Step3: Without Redaction:

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree displays various database objects like EQUIPMENT, EQUIPMENT_LIST, MATERIAL, MATERIAL_LIST, PROJECT, PROJECT_TEAM, STAFF, and VENDOR. A sequence named SSN_SEQUENCE is selected. The central workspace contains a PL/SQL script:

```

1 declare
2   i number(10);
3   count1 number(4);
4   curval number(10);
5   staffid integer;
6
7   cursor cur_count IS select count(*)from staff;
8
9 BEGIN
10   open cur_count;
11   fetch cur_count into count1;
12   close cur_count;
13
14   curval :=325487458;
15   staffid :=3000;
16   FOR i in 0..count1  LOOP
17     update staff set ssn=curval+i where staff_id=staffid+i;
18   END LOOP;
19   END;
20 /
21 select * from staff;
22 --select distinct count(ssn) from staff;

```

The bottom pane shows the results of the query, titled "Query Result 1". It lists 50 rows of staff information, including Staff ID, Name, Contact, Designation, Wages, and SSN. The SSN values are generated by adding the staff ID to the initial value of 325487458.

STAFF_ID	STAFF_NAME	STAFF_CONTACT	DESIGNATION	WAGES	SSN
1	3001 Ardine	8647354672	Operator	1000	325487459
2	3002 Benson	4364968612	Operator	1000	325487460
3	3003 Agnella	3516932814	Operator	1000	325487461
4	3004 Perice	8783056574	Operator	1000	325487462
5	3005 Remus	5123639224	Operator	1000	325487463
6	3006 Eb	7583123901	Operator	1000	325487464
7	3007 Juditha	5759280574	Operator	1000	325487465
8	3008 Roldan	7742584175	Operator	1000	325487466
9	3009 Jamaal	6021820726	Operator	1000	325487467

Step 4: with Data Redaction

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

Script Output x Query Result x
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	Gordur	6309935196	Operator	1000 VVV-VV-69	

8. DBA Scripts:

➤ Fetch Current Running Script:

Displays the SQL statements for currently running processes.

```
1  SET LINESIZE 500
2  SET PAGESIZE 1000
3  SET FEEDBACK OFF
4
5  SELECT s.sid,
6         s.status "Status",
7         p.spid "Process",
8         s.schemaname "Schema Name",
9         s.osuser "OS User",
10        Substr(a.sql_text,1,120) "SQL Text",
11        s.program "Program"
12   FROM v$session s,
13        v$sqlarea a,
14        v$process p
15  WHERE s.sql_hash_value = a.hash_value (+)
16  AND s.sql_address      = a.address (+)
17  AND s.paddr            = p.addr;
18
19  SET PAGESIZE 14
20  SET FEEDBACK ON
```

The screenshot shows the Oracle SQL Developer interface. The top navigation bar has tabs for Start Page, Achyuth Reddy Kota.sql, Achyuth Reddy Kota, PROJECT, and Achyuth Reddy Kota_1. The status bar indicates 2.1319997 seconds. The main area has two tabs: Worksheet and Query Builder. The Worksheet tab contains the SQL code from the previous slide. The Query Result tab shows the output of the query, which is a table with columns SID, Status, Process, Schema Name, OS User, and SQL Text. The table lists various sessions with their details. The bottom part of the interface shows the Script Output tab with the executed SQL and the Query Result tab with the fetched data.

SID	Status	Process	Schema Name	OS User	SQL Text
49	85 INACTIVE	2124	DB2GREEN	jamesbond	(null)
50	33 INACTIVE	764	DB207	ramreddy	(null)
51	68 INACTIVE	2432	MIMOTO	ram	(null)
52	55 INACTIVE	4024	DB247	yadav	(null)
53	58 INACTIVE	2444	DB226	prave	(null)
54	111 INACTIVE	1872	DB2GREEN	bsoni	(null)
55	70 INACTIVE	860	DB287	rahul	/* + NO_PARALLEL */SELECT ROWID "ROWID", ORA_ROWSCN "ORA_ROWSCN", APPT_ID APPT_ID, PROF_EM
56	102 INACTIVE	2772	DB247	raush	select * from IMUSICAPP_ALBUMS
57	95 ACTIVE	3388	DB281	U	SELECT s.sid, s.status "Status", p.spid "Process", s.schemaname "Sche
58	67 INACTIVE	600	TEST2	RC	(null)
59	107 INACTIVE	2408	DB205	Mayurdhvaj	(null)
60	74 INACTIVE	564	DB281	U	(null)
61	109 INACTIVE	1164	DBERNDT	dberndt	SELECT users.user_id, distances.dist / (SQRT(my.norm) * SQRT(users.norm)) AS score

➤ 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;
```

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, there are tabs for Start Page, Achyuth Reddy Kota.sql, Achyuth Reddy Kota, PROJECT, and Achyuth Reddy Kota_1. Below the tabs, the Worksheet tab is active, showing the following SQL query:

```
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';
```

In the bottom right corner of the worksheet area, it says "All Rows Fetched: 21 in 0.057 seconds". Below the worksheet, the Query Result tab is active, displaying a table with 21 rows of data:

	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.

```
1 COLUMN size_for_estimate      FORMAT 999,999,999,999 heading 'Cache Size (MB)'
2 COLUMN buffers_for_estimate   FORMAT 999,999,999 heading 'Buffers'
3 COLUMN estd_physical_read_factor FORMAT 999.90 heading 'Estd Phys|Read Factor'
4 COLUMN estd_physical_reads    FORMAT 999,999,999,999 heading 'Estd Phys| Reads'
5
6 SELECT size_for_estimate,
7     buffers_for_estimate,
8     estd_physical_read_factor,
9     estd_physical_reads
10    FROM v$db_cache_advice
11   WHERE name      = 'DEFAULT'
12   AND block_size  = (SELECT value
13                         FROM v$parameter
14                       WHERE name = 'db_block_size')
15   AND advice_status = 'ON';
```

The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes tabs for Start Page, Achyuth Reddy Kota.sql, Achyuth Reddy Kota, PROJECT, and Achyuth Reddy Kota_1. The toolbar below has various icons for file operations. The main area is divided into two panes: Worksheet and Query Builder. The Worksheet pane contains the SQL script from the previous code block. The Query Result pane at the bottom shows the output of the query, which is a table with 10 rows of data. The table has columns: SIZE_FOR_ESTIMATE, BUFFERS_FOR_ESTIMATE, ESTD_PHYSICAL_READ_FACTOR, and ESTD_PHYSICAL_READS. The data is as follows:

	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

```
1  SET LINESIZE 200
2
3  COLUMN object_owner FORMAT A20
4  COLUMN object_name FORMAT A30
5  COLUMN policy_name FORMAT A30
6  COLUMN expression FORMAT A30
7  COLUMN policy_description FORMAT A20
8
9  SELECT object_owner,
10       object_name,
11       policy_name,
12       expression,
13       enable,
14       policy_description
15  FROM redaction_policies
16  ORDER BY 1, 2, 3;
```

The screenshot shows the Oracle SQL Developer interface. The top menu bar has tabs for '...sql', 'PROJECT', and several database connections. The main window has tabs for 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying the SQL query from the previous code block. Below the query, the results are shown in a table titled 'SQL'. The table has columns: OBJECT_OWNER, OBJECT_NAME, POLICY_NAME, EXPRESSION, ENABLE, and POLICY_DESCRIPTION. The data is as follows:

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

```
1  SET PAGESIZE 0
2  SET FEEDBACK OFF
3  SET VERIFY OFF
4
5  SPOOL temp.sql
6
7  SELECT 'GRANT DELETE ON "' || u.table_name || '"' TO &1;
8  FROM  user_tables u
9  WHERE NOT EXISTS (SELECT '1'
10                    FROM  all_tab_privs a
11                    WHERE a.grantee   = UPPER('&1')
12                    AND   a.privilege = 'DELETE'
13                    AND   a.table_name = u.table_name);
14
15 SPOOL OFF
16
17 @temp.sql
18
19 SET PAGESIZE 14
20 SET FEEDBACK ON
21 SET VERIFY ON
```

Z

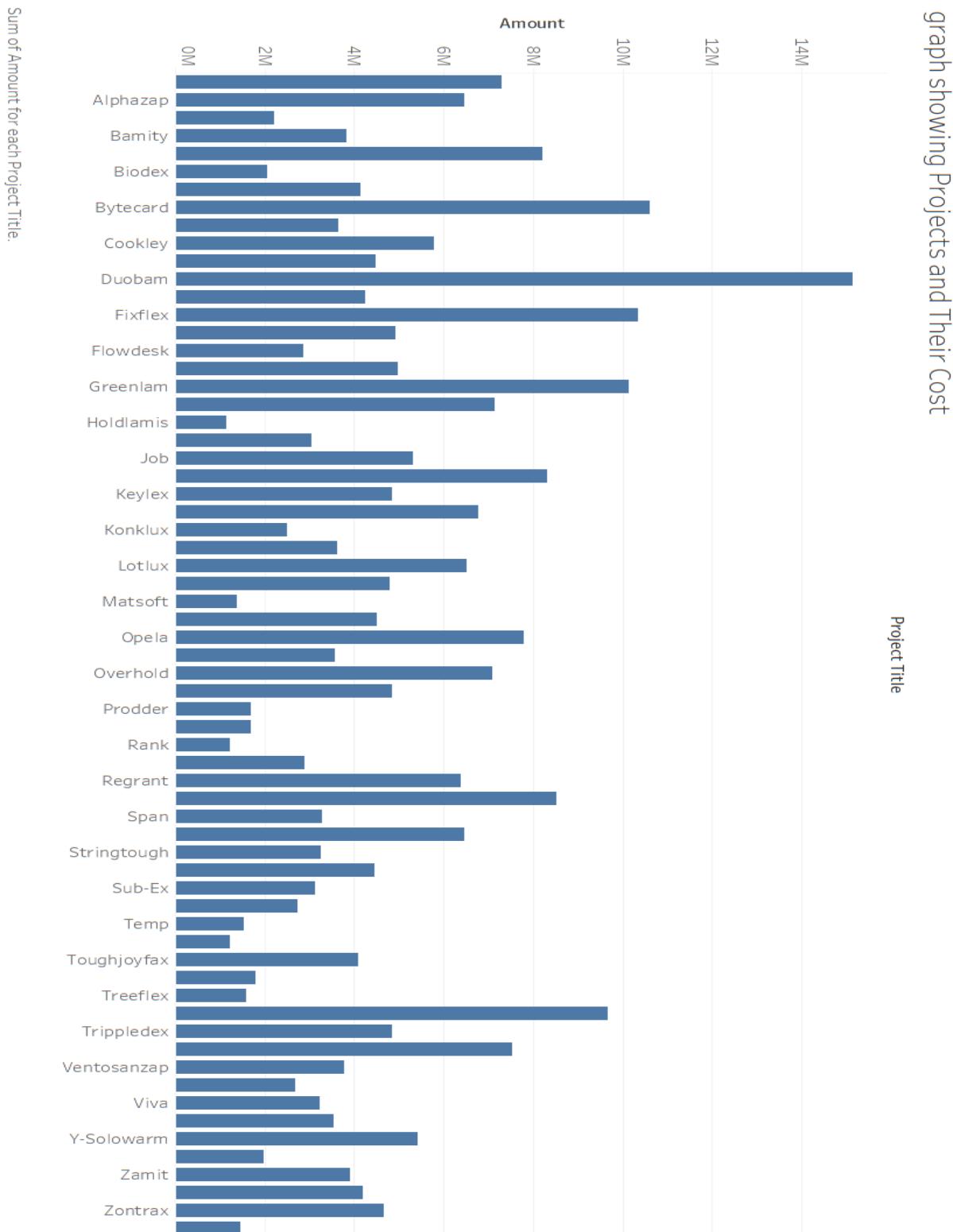
The screenshot shows the Oracle SQL Developer interface. The top menu bar has tabs for 'PROJECT' and two sessions named 'Achyuth Reddy Kota_1' and 'Achyuth Reddy Kota_2'. The toolbar includes icons for running scripts, saving, and zooming. The main window has tabs for 'Worksheet' and 'Query Builder', with 'Worksheet' selected. The worksheet area contains the SQL script provided above. Below the script, the command 'SPOOL OFF' is visible. The bottom part of the interface shows the 'Script Output' tab, which displays the results of the executed grants. The output table has a header row with columns 'GRANTDELETEON', 'TABLE_NAME', and 'TOU'. The data rows show 11 grants for various tables: 'CLIENT', 'CLIENT_TRANSACTIONS', 'EQUIPMENT', 'EQUIPMENT_LIST', 'MATERIAL', 'MATERIAL_LIST', 'PROJECT', 'PROJECT_TEAM', 'STAFF', 'STAFFREDACTION', and 'TEST_MATERIAL'. Each grant is preceded by a number from 1 to 11.

GRANTDELETEON	TABLE_NAME	TOU
1	GRANT DELETE ON "CLIENT" TO U;	
2	GRANT DELETE ON "CLIENT_TRANSACTIONS" TO U;	
3	GRANT DELETE ON "EQUIPMENT" TO U;	
4	GRANT DELETE ON "EQUIPMENT_LIST" TO U;	
5	GRANT DELETE ON "MATERIAL" TO U;	
6	GRANT DELETE ON "MATERIAL_LIST" TO U;	
7	GRANT DELETE ON "PROJECT" TO U;	
8	GRANT DELETE ON "PROJECT_TEAM" TO U;	
9	GRANT DELETE ON "STAFF" TO U;	
10	GRANT DELETE ON "STAFFREDACTION" TO U;	
11	GRANT DELETE ON "TEST_MATERIAL" TO U;	

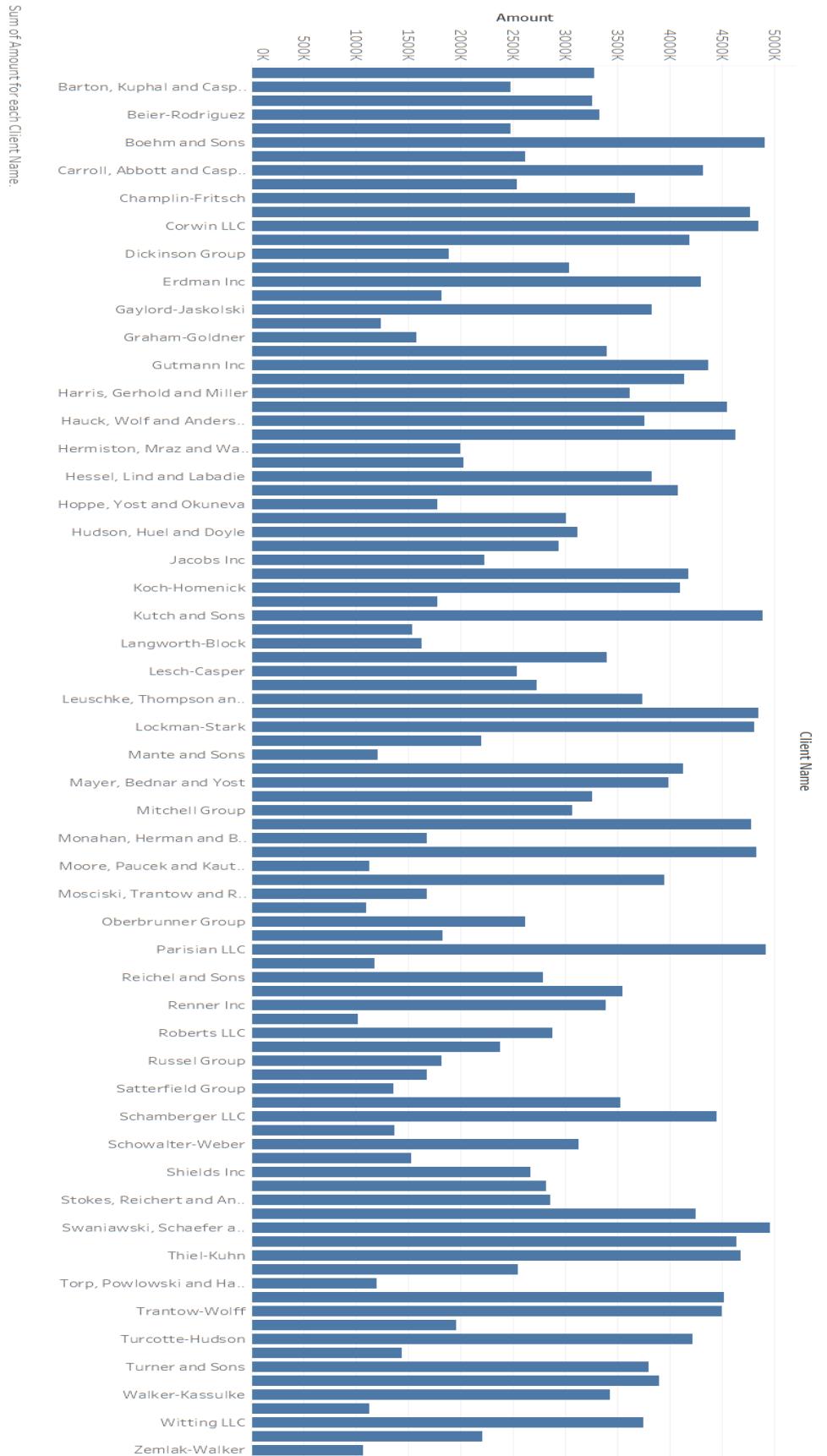
9. Data Visualization

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

➤ Budget Distribution of the Projects



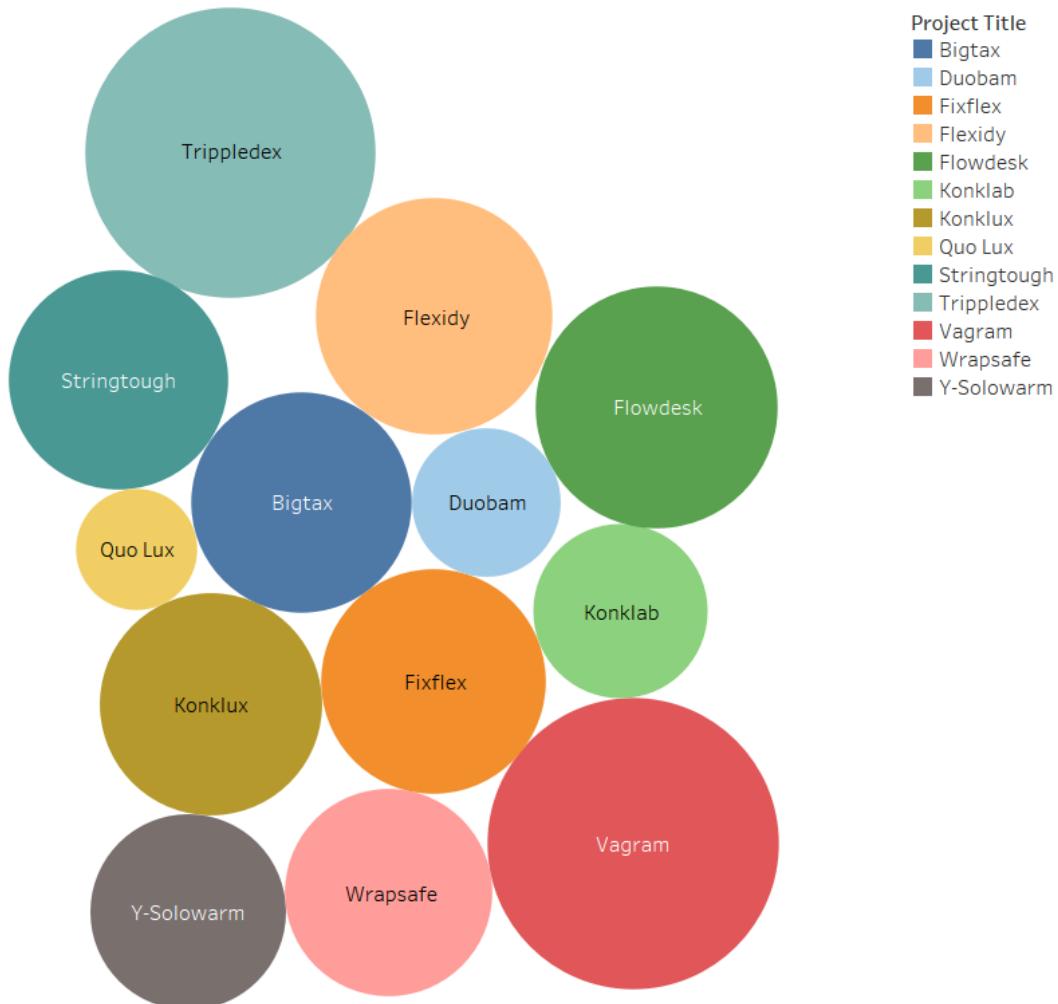
➤ Budget Distributions of the Clients



➤ **Visualiation of the Projects' Duration**

➤

graph showing Projects and Their Duration



10. Optimizer - Performance Tuning

➤ Query without Optimizer :

```
1 Select eq_list, equipment_eq_id, material_list, material_material_id, project_title,
2     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';
```

The screenshot shows the Oracle SQL Developer interface. The top navigation bar has tabs for Start Page, Achyuth Reddy Kota.sql, Achyuth Reddy Kota, PROJECT, Achyuth Reddy Kota_1, Achyuth Reddy Kota_2, and Achyuth Reddy Kota_3... The main area is divided into two panes. The left pane contains the SQL query:--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';

```
The right pane is currently empty. Below these panes is a toolbar with various icons. At the bottom, there are tabs for Query Result, Query Result 1, and Query Result 2, followed by a SQL tab and a note that 50 rows were fetched in 0.144 seconds. The Query Result tab displays a table with the following data:
```

EQ_LIST	EQUIPMENT_EQ_ID	MATERIAL_LIST	MATERIAL_MATERIAL_ID	PROJECT_TITLE	PROJECT_LOCATION	START_DATE	END_DATE	PROJECT_STATUS
1	343480141	111681	565680141	S110005 Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
2	343480141	111334	565680141	S110005 Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
3	343480141	111275	565680141	S110005 Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
4	343480141	111195	565680141	S110005 Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
5	343480141	111023	565680141	S110005 Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
6	343480141	111681	565680141	S110067 Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
7	343480141	111334	565680141	S110067 Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
8	343480141	111275	565680141	S110067 Bigtax	Los Pinos	03-MAR-17	08-OCT-18	ACTIVE
9	343480141	111195	565680141	S110067 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

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, there are several tabs: Start Page, Achyuth Reddy Kota.sql, Achyuth Reddy Kota, PROJECT, Achyuth Reddy Kota_1, Achyuth Reddy Kota_2, and Achyuth Reddy Kota_3... The PROJECT tab is currently active.

In the main area, there is a Worksheet tab and a Query Builder tab. The Worksheet tab contains the following SQL code:

```
--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';
```

Below the code, the status message "All Rows Fetched: 330 in 0.367 seconds" is displayed.

The Query Result tab shows the fetched data in a grid format:

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

```
1 FIRST_ROWS(10) optimiser
2
3 Select /*+ FIRST_ROWS(10) */ eq_list, equipment_eq_id, material_list,
4 | material_material_id, project_title, project_location, start_date, end_date, project_status
5 from equipment_list inner join project
6 on equipment_list.project_project_id = project.project_id
7 inner join material_list
8 on project.project_id = material_list.project_project_id
9 where project_title = 'Bigtax';
```

The screenshot shows the Oracle SQL Developer interface. The top navigation bar has tabs for Start Page, Achyuth Reddy.Kota.sql, Achyuth Reddy Kota, PROJECT, Achyuth Reddy Kota_1, Achyuth Reddy Kota_2, and Achyuth Reddy Kota_3... The main area is divided into two panes: a Worksheet pane and a Query Builder pane. The Worksheet pane contains the SQL query shown above. The Query Builder pane shows the query plan with nodes like 'FIRST_ROWS(10) optimiser' and various table and index scans. Below these panes is a toolbar with icons for running queries, saving, and connecting. The bottom section is a Query Result pane titled 'Query Result' which displays a table of 50 rows. The table has columns: EQ_LIST, EQUIPMENT_EQ_ID, MATERIAL_LIST, MATERIAL_MATERIAL_ID, PROJECT_TITLE, PROJECT_LOCATION, START_DATE, END_DATE, and PROJECT_STATUS. The data shows multiple entries for 'Bigtax' projects.

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

The screenshot shows the Oracle SQL Developer interface. The top navigation bar has tabs for Start Page, Achyuth Reddy Kota.sql, Achyuth Reddy Kota, PROJECT, Achyuth Reddy Kota_1, Achyuth Reddy Kota_2, and Achyuth Reddy Kota_3... The main area is a Worksheet tab where a SQL query is being run. The query uses the FIRST_ROWS(10) optimizer hint:

```
-- FIRST_ROWS(10) optimiser
Select /*+ FIRST_ROWS(10) */ eq_list.equipment_eq_id, material_list.material_material_id, project.project_title, project.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';
```

The results are displayed in a grid titled "Query Result". The grid has columns: EQ_LIST, EQUIPMENT_EQ_ID, MATERIAL_LIST, MATERIAL_MATERIAL_ID, PROJECT_TITLE, PROJECT_LOCATION, START_DATE, END_DATE, and PROJECT_STATUS. The data shows 330 rows, all of which are marked as ACTIVE.

	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';
```

The screenshot shows the Oracle SQL Developer interface. The top navigation bar has tabs for Start Page, Achyuth Reddy Kota.sql, Achyuth Reddy Kota, PROJECT, Achyuth Reddy Kota_1, Achyuth Reddy Kota_2, and Achyuth Reddy Kota_3. The main area is a Worksheet tab with a Query Builder icon. The query editor contains the following code:

```
-- All ROWS optimiser mode
Select /*+ All_ROWS */ eq_list, equipment_eq_id, material_list, material_material_id,
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';
```

Below the code, the message "Fetched 50 rows in 0.037 seconds" is displayed. A result grid shows the fetched data:

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

Start Page x Achyuth Reddy Kota.sql x Achyuth Reddy Kota x PROJECT x Achyuth Reddy Kota_1 x Achyuth Reddy Kota_2 x Achyuth Reddy Kota_3...

SQL Worksheet History

Worksheet Query Builder

```
-- All ROWS optimiser mode
Select /*+ All_ROWS */ eq_list.equipment_eq_id, material_list.material_material_id, project.project_title, project.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';
```

Query Result x | Query Result 2 x | Query Result 3 x | Query Result 4 x | Query Result 5 x | Query Result 6 x | Query Result 7 x | Quer... x | x

SQL | All Rows Fetched: 330 in 0.261 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

All Rows were fetched in 0.261 seconds