

DATABASE SPECIFICATIONS

Real Estate Development Database Management System

Tools Used: Oracle SQL

Techniques Used: Relational database design (define purpose, ERD, tables schemas, keys, normalization, etc.); SQL to create, alter, and populate tables, manipulate data, query/retrieve data; Relational algebra query

Kimberly Healy
healy.kim@gmx.us

DOCUMENT CONTROL

Work carried out by:

Name	Email Address	Other
Kimberly Healy	healy.kim@gmx.us	

Revision Sheet

Release No.	Date	Revision Description

DATABASE SPECIFICATIONS

TABLE OF CONTENTS

Document Control

Work carried out by:	1
Revision Sheet.....	1

1. GENERAL INFORMATION

Purpose.....	3
System Overview.....	3

2. ARCHITECTURE DESIGN

Assumptions and Constraints.....	5
Entity Relationship Diagram.....	5
Data Dictionary.....	6

3. DATABASE IDENTIFICATION AND DESCRIPTION

Assumptions and Constraints	11
Naming Conventions.....	11
Tables Schemas.....	11

4. DATABASE NORMALIZATION

Assumptions and Constraints.....	26
Tables Normal Form.....	26

5. SQL QUERY

Query 1.....	32
Query 2.....	32
Query 3.....	33
Query 4.....	33
Query 5.....	33
Query 6.....	34
Query 7.....	34
Query 8.....	35

1. GENERAL INFORMATION

Purpose

The purpose of this database is to provide a medium sized east-cost real estate development firm, Suedan, the storage, access, and ability to manipulate necessary data that accumulates over the process of real estate development. The process of real estate development at Suedan is as follows:

1. Search for real estate deals;
2. Conduct due diligence on potential properties;
3. Buy real estate property;
4. Develop a new building design;
5. Hire builders to construct the new building design;
6. Choose to maintain the developed building and rent out space OR sell the developed building.

System Overview

- System name or title: Real Estate Development Database Management System
- Core requirements:
 - Suedan is well-connected with real estate agent companies that help them find and facilitate purchases of real estate. Suedan stores every real estate agent company's contact data. There may be one or multiple contacts at the company that Suedan can contact directly. Suedan stores their name(s) in the database. Every company is identified in the database by a **AGENTid** number.
 - If there is a property of interest to Suedan, they collect data on the property. The property is categorized into types: Building, Lot, or both Building and Lot.
 - Attributes collected about the property as a whole are identified in the database by a **REid** number. These attributes include the address, type, physical description, date and price of last ownership turnover, property taxes, and seller's asking price. Each property may be associated with a real estate agent.
 - Local attributes collected about the property's building(s) include size in square feet, number of stories, date building was built, and a physical description.
 - Local attributes collected about the property's lot include size in acres and physical description.
 - Building design & creation: Architecture & Construction
 - Suedan hires architectural companies to redesign the undeveloped property. Suedan stores every architect's contact data along with one or multiple main contacts. Every company is identified in the database by a **ARCHid** number.
 - Suedan hires construction firms to build the new design. Suedan stores every construction company's contact data along with one or multiple main contacts. Every company is identified in the database by a **CONSTRid** number.
 - Suedan keeps data on the progress of each property development project, which is identified by a **PROJECTid** number. The attributes needed for every project include the following: historical data about the property (REid), buying details (price, date), date of construction completion, total development costs, current valuation of the property, profit made from the development, and current status of the property (Own, Rent, For Sale, Sold). Additionally, the construction company,

architectural company, and employees who worked on the project need to be linked to the unique project.

- Suedan employees are identified by their **EMPLOYid** numbers. Attributes stored about each employee include name, sex, date of birth, home address, phone number, and EMPLOYid of the employee's manager. Employees are assigned to manage projects. The number of projects can range from zero to many.

- Table Standards

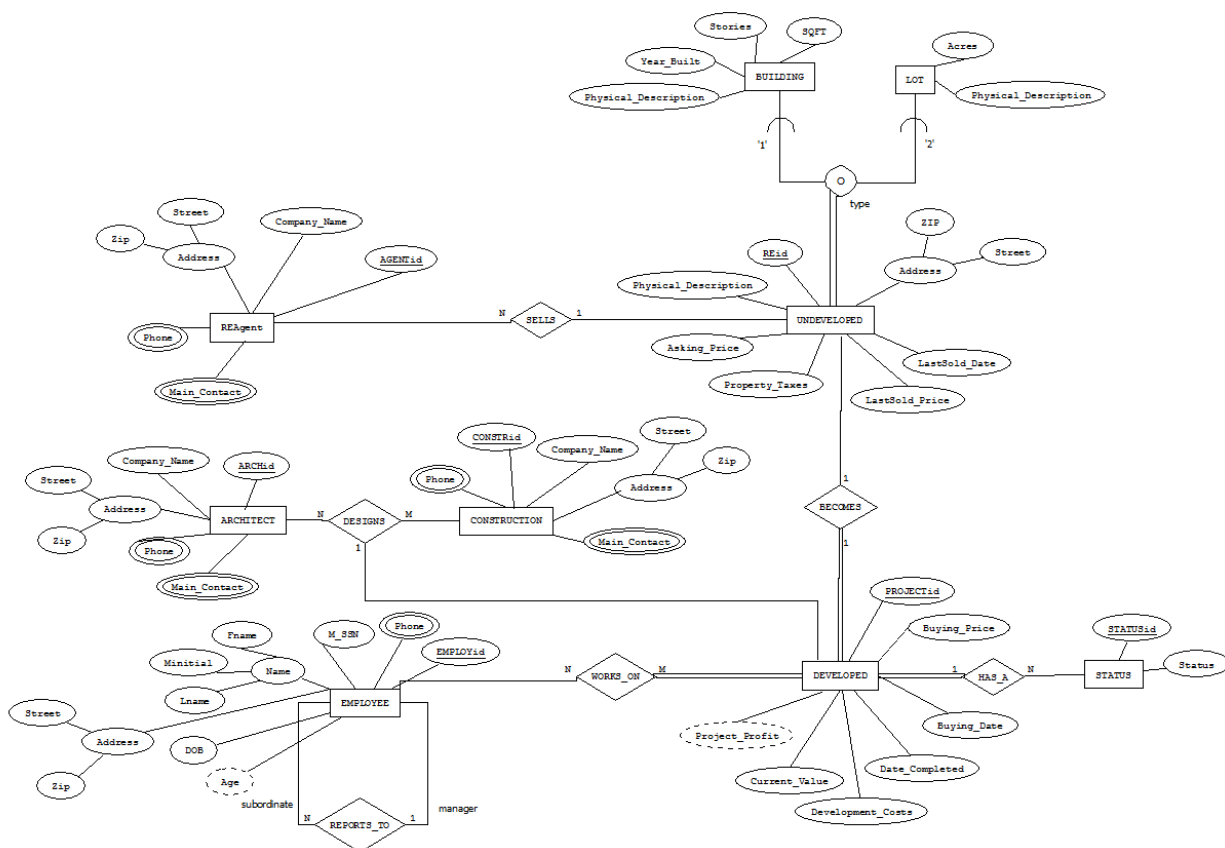
- All the tables will be normalized up to the third normal form in order to control redundancy.
- Primary keys such as the element IDs will be used to uniquely identify the records.
- Foreign key constraints will be used to maintain referential integrity throughout the database.
- Table names should be descriptive and reflect the table's content and function.
- Each column/element name must be unique within the table.
- Abbreviations should be standard and logical.

2. ARCHITECTURE DESIGN

Assumptions and Constraints

- Each entity has multiple attributes, which can be composite or multivalued. Key constraint on attributes holds and is denoted by an underlined attribute.
- The following attributes are derived: Age, Project_Profit
- The following entity relationships are possible: one to one; one to many; many to many. Cardinality ratios are expressed in the ERD as 1:1, 1:N, N:M.
- Participation in an entity relationship can be total (double line) or partial (single line).
- Recursive relationships must specify role names of participating entities.
- Superclass/subclass relationships are predicate-defined. Superclass/subclass relationships abide by the type inheritance constraint. Additionally, every entity in a superclass can either be a member of more than one subclass or at most one, denoted by 'o' and 'd' respectively.
- *The ZIP_CODE entity is excluded from the ER Diagram and is replaced by a Zip attribute. This entity is in a relationship with any entity with an Address attribute (UNDEVELOPED, REAGENT, ARCHITECT, CONSTRUCTION).*

Entity Relationship Diagram



Data Dictionary

Zip_Code Table

Description: This table is a reference for any Address attributes and contains Zip codes for within the United States of America.

Variable	Variable name	Variable type	Values	notes
Zip code	ZIP	Numeric	00000-99999	Primary key
City	City	String	Malvern	
State	STATE	String	All 50 states i.e. PA	

Undeveloped Table

Description: Suedan stores information about potential real estate that they are interested in buying. This table also provides historical information about properties that they choose to develop.

Variable	Variable name	Variable type	Values	notes
Real Estate ID number	REID	Numeric	0000-9999	Primary key
Property street address	STREET	String	90 Main Street	
Property zip code	ZIP	Numeric	00000-99999	
Type	TYPE	Numeric	1 = Building 2 = Lot	Multivalued attribute
Physical Description in words	PHYSICAL_DESCRIPTION	String		
Date property was last sold	LASTSOLD_DATE	mm/dd/yyyy	1-12/1-31/1900-2017	
Price property was last sold	LASTSOLD_PRICE	Numeric	000000000-999999999	
Sale price of property	ASKING_PRICE	Numeric	000000000-999999999	
Property taxes	PROPERTY_TAXES	Numeric	000000000-999999999	

Building Table <i>Description:</i> Undeveloped Real Estate can be categorized into 'Building' or 'Lot' type.				
Variable	Variable name	Variable type	Values	notes
Real Estate ID number	REID	Numeric	0000-9999	Primary key
Building size: square feet	SQFT	Numeric	000000-999999	
Building size: stories	STORIES	Numeric	000-150	
Year building was built	YEAR_BUILT	Numeric	1800-2017	
Physical Description in words	PHYSICAL_DESCRIPTION	String		

Lot Table <i>Description:</i> Undeveloped Real Estate can be categorized into 'Building' or 'Lot' type.				
Variable	Variable name	Variable type	Values	notes
Real Estate ID number	REID	Numeric	0000-9999	Primary key
Lot size: acres	ACRES	Numeric	000000-999999	
Physical Description in words	PHYSICAL_DESCRIPTION	String		

REAgent Table <i>Description:</i> Suedan stores the contact information of all real estate companies that they do business with. A main contact or multiple main contacts may be identified.				
Variable	Variable name	Variable type	Values	notes
Company ID number	AGENTID	Numeric	0000-9999	Primary key
Company name	COMPANY_NAME	String		

Company zip code	ZIP	Numeric	00000-99999	Foreign key
Company street address	ADDRESS	String	10 Run Lane	
Phone Number	PHONE	Numeric	000-000-0000	Multivalued attribute
Main contact	MAIN_CONTACT	String	Lname, Fname	Multivalued attribute

Status Table				
Description: Contains descriptors of the status of Developed Real Estate.				
Variable	Variable name	Variable type	Values	notes
Status ID number	STATUSID	Numeric	00-99	Primary key
Status of property	STATUS	Numeric	0 = In development 1 = Own 2 = Rent 3 = For sale 4 = Sold	

Developed Table				
Description: Suedan chooses which Undeveloped Real Estate properties to purchase. Developed Real Estate contains the data for each project.				
Variable	Variable name	Variable type	Values	notes
Project ID number	PROJECTID	Numeric	0000-9999	Primary key
Buying price of property	BUY_PRICE	Numeric	000000000-999999999	
Date of purchase	BUY_DATE	mm/dd/yyyy	1-12/1-31/1900-2017	
Date of development competition	DATE_COMPLETED	mm/dd/yyyy	1-12/1-31/1900-2017	
Total cost of development	DEVELOP_COSTS	Numeric	000000000-999999999	

Current property valuation	CURRENT_VALUE	Numeric	000000000-999999999	
Profit from project	PROJECT_PROFIT	Numeric	000000000-999999999	Derived attribute from BUY_PRICE, DEVELOP_COSTS and CURRENT_VALUE
Status of property	STATUSID	Numeric	0 = In development 1 = Own 2 = Rent 3 = For sale 4 = Sold	Foreign key

Architect Table

Description: Suedan stores the contact information of all Architectural companies that they do business with. A main contact or multiple main contacts may be identified.

Variable	Variable name	Variable type	Values	notes
Company ID number	ARCHID	Numeric	0000-9999	Primary key
Company name	COMPANY_NAME	String		
Company zip code	ZIP	Numeric	00000-99999	Foreign key
Company street address	ADDRESS	String	10 Run Lane	
Phone Number	PHONE	Numeric	000-000-0000	Multivalued attribute
Main contact	MAIN_CONTACT	String	Lname, Fname	Multivalued attribute

Construction Table				
Description: Suedan stores the contact information of all Construction companies that they do business with. A main contact or multiple main contacts may be identified.				
Variable	Variable name	Variable type	Values	notes
Company ID number	CONSTID	Numeric	0000-9999	Primary key
Company name	COMPANY_NAME	String		
Company zip code	ZIP	Numeric	00000-99999	Foreign key
Company street address	ADDRESS	Numeric, String	Street, City, State, Zip	Composite attribute
Phone Number	PHONE	Numeric	000-000-0000	Multivalued attribute
Main contact	MAIN_CONTACT	String	Lname, Fname	Multivalued attribute

Employee Table				
Description: An employee is a person that works for Suedan.				
Variable	Variable name	Variable type	Values	notes
Employee ID number	EMPLOYID	Numeric	0000-9999	Primary key
Employee name	NAME	String	Lname, Fname, Minitia	Composite attribute
Birthday	DOB	mm/dd/yyyy	1-12/1-31/1900-2010	
Age in years	AGE	Numeric	000-100	Derived attribute from DOB and current date
Home zip code	ZIP	Numeric	00000-99999	Foreign key
Home street address	STREET	String	10 Long Lane	
Phone Number	PHONE	Numeric	000-000-0000	Multivalued attribute
Manager's employee ID	M_EMPLOYID	Numeric	0000-9999	

3. DATABASE IDENTIFICATION AND DESCRIPTION

Assumptions and Constraints

Naming Conventions

- Abbreviations used in the tables schemas are as follows:
 - RE stands for Real Estate
 - ARCH stands for Architect
 - CONSTR stands for Construction
 - EMPLOY stands for Employee
- Undeveloped implies Undeveloped Real Estate
- Developed implies Developed Real Estate

Tables schemas

Name of the table	ZIP_CODE		
Description	This table is a reference for any Address attributes.		
Attribute	Description	Type	<i>Examples of values</i>
ZIP	Zip code	Number	19355
CITY	Address	String	Malvern
STATE	Address	String	PA
Primary Key	ZIP		
Foreign Keys	None		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.ZIP_CODE (ZIP NUMBER(5) NOT NULL, CITY VARCHAR2(30) NOT NULL, STATE CHAR(2) NOT NULL, PRIMARY KEY(ZIP)); INSERT INTO ZIP_CODE VALUES('19355', 'MALVERN', 'PA'); INSERT INTO ZIP_CODE VALUES('02115', 'BOSTON', 'MA'); INSERT INTO ZIP_CODE VALUES('20016', 'WASHINGTON', 'DC'); INSERT INTO ZIP_CODE VALUES('29910', 'BLUFFTON', 'SC'); INSERT INTO ZIP_CODE VALUES('11705', 'BAYPORT', 'NY'); INSERT INTO ZIP_CODE VALUES('06335', 'MYSTIC', 'CT');</pre>		
Count of records in the table	6		

Name of the table	UNDEVELOPED		
Description	This table describes information about potential real estate that Suedan is interested in buying. This table is also a historical reference for real estate that Suedan chooses to develop.		
Attribute	Description	Type	<i>Examples of values</i>
REID	Identifier of a real estate property	Number	Between 0 and 9999
ZIP	Address	Number	19355
STREET	Address	String	8 King Street
TYPE	Building or Lot property	Number	1 = Building 2 = Lot
PHYSICAL_DESCRIPTION	Physical description of property in employee's words	String	Located near main pedestrian walkway. Lot contains 1 tear-down building in NE corner.
LASTSOLD_DATE	Date property was last sold to current owner	Date	Between 1900-01-01 to current date
LASTSOLD_PRICE	Price property was last sold to current owner	Number	1000000
ASKING_PRICE	Current sale price of property	Number	2000000
PROPERTY_TAXES	Total taxes on property	Number	100000
Primary Key	REID		
Candidate Key	(ZIP, Street)		
Foreign Keys	ZIP		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.UNDEVELOPED (REID NUMBER(4) NOT NULL, ZIP NUMBER(5) NOT NULL, STREET VARCHAR2(50) NOT NULL, TYPE INT NOT NULL CHECK(TYPE > 0 AND TYPE < 3), PHYSICAL_DESCRIPTION VARCHAR2(4000), LASTSOLD_DATE DATE, LASTSOLD_PRICE NUMBER(9,2), ASKING_PRICE NUMBER(9,2), PROPERTY_TAXES NUMBER(9,2), PRIMARY KEY(REID), FOREIGN KEY(ZIP) REFERENCES ZIP_CODE(ZIP));</pre>		

	INSERT INTO UNDEVELOPED VALUES('1001', '02115', '9088 GABRIELLA WAY', '1', 'CORNER LOT. CURRENTLY OCCUPIED BY 7-11 CONVENIENCE STORE. NEIGHBORING A PARKING GARAGE AND FENWAY STADIUM.', date '2001-01-01', '506000', '900000', '100000'); INSERT INTO UNDEVELOPED VALUES('1002', '02115', '45 SYMPHONY RD', '1', 'NEIGHBORING A CHURCH AND A FREEWAY ENTRANCE.', date '2005-11-11', '800000', '1400000', '200000'); INSERT INTO UNDEVELOPED VALUES('1003', '19355', '8 KING ROAD', '2', 'EMPTY LOT ON MAIN PEDESTRIAN STREET. COMMERCIAL ZONED. CURRENT OWNER LEASING LAND.', date '2015-01-31', '205000', '400000', '10000'); INSERT INTO UNDEVELOPED VALUES('1004', '11705', '600 FRAZER LANE', '2', 'OWNED BY COMMERCIAL OFFICE PARK WITH 5 HIGHRISES.', date '1980-03-12', '800000', '2000000', '950000');
Count of records in the table	4

Name of the table	BUILDING		
Description	This table is a subclass of Undeveloped and describes the Building(s) located on Undeveloped properties.		
Attribute	Description	Type	<i>Examples of values</i>
REID	Identifier of a real estate property	Number	Between 0 and 9999
SQFT	Square feet of building	Number	Between 0 and 999999
STORIES	Number of stories of the building	Number	Between 0 and 150
YEAR_BUILT	Year the building was built	Number	1980
PHYSICAL_DESCRIPTION	Description in words of the building	String	10 rooms/2 bath
Primary Key	REID		
Foreign Keys	none		
SQL Code	CREATE TABLE USERINSC521SU17_KZH75.BUILDING (REID NUMBER(4) NOT NULL, SQFT NUMBER(6), STORIES NUMBER(3), YEAR_BUILT NUMBER(4) CHECK(YEAR_BUILT > 1800 AND YEAR_BUILT < 2017), PHYSICAL_DESCRIPTION VARCHAR2(4000), PRIMARY KEY(REID));		

	INSERT INTO BUILDING VALUES('1001', '2050', '2', '1953', 'RENOVATED IN 2004. 5 ROOM/2 BATH.');
	INSERT INTO BUILDING VALUES('1002', '3100', '3', '1998', '6 ROOM/3 BATH. CEDAR FLOOR THROUGHOUT.');
Count of records in the table	2

Name of the table	LOT		
Description	This table is a subclass of Undeveloped and describes the Lot(s) located on Undeveloped properties.		
Attribute	Description	Type	<i>Examples of values</i>
REID	Identifier of a real estate property	Number	Between 0 and 9999
ACRES	Number of acres of the lot	Number	Between 0 and 999999
PHYSICAL_DESCRIPTION	Description in words of the lot	String	Gravel, tear-down building located in NW corner.
Primary Key	REID		
Foreign Keys	none		
SQL Code	CREATE TABLE USERINSC521SU17_KZH75.LOT (REID NUMBER(4) NOT NULL, ACRES NUMBER(3,2), PHYSICAL_DESCRIPTION VARCHAR2(4000), PRIMARY KEY(REID)); INSERT INTO LOT VALUES('1003', '0.85', 'GRAVEL DRIVE.');		
Count of records in the table	2		

Name of the table	REAGENT		
Description	This table provides contact information about the real estate companies that Suedan does business with.		
Attribute	Description	Type	<i>Examples of values</i>
AGENTID	Identifier of a real estate agent company	Number	Between 0 and 9999
REID	Identifier of a real estate property	Number	Between 0 and 9999

COMPANY_NAME	Name of company	String	Bluff Real Estate
ZIP	Address	Number	19355
STREET	Address	String	90 Needle Ln
Primary Key	AGENTID, REID		
Foreign Keys	ZIP		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.REAGENT (AGENTID NUMBER(4) NOT NULL, REID NUMBER (4) NOT NULL, COMPANY_NAME VARCHAR2(30) NOT NULL, ZIP NUMBER(5), STREET VARCHAR2(50), PRIMARY KEY(AGENTID, REID), FOREIGN KEY (ZIP) REFERENCES ZIP_CODE(ZIP), UNIQUE(AGENTID)); INSERT INTO REAGENT VALUES ('0001', '1004', 'REMAXX OF LONG ISLAND', '11705', '789 LONG ISLAND AVE'); INSERT INTO REAGENT VALUES ('0002', '1002', 'REMAXX OF CONNETICUT', '21122', '345 MYSTIC AVE');</pre>		
Count of records in the table	2		

Name of the table	REAGENT_PHONE		
Description	This table provides the multiple phone numbers of each real estate company.		
Attribute	Description	Type	Examples of values
AGENTID	Identifier of a real estate agent company	Number	Between 0 and 9999
PHONE	Company's phone number(s)	Number	212-666-0000
Primary Key	(AGENTID, PHONE)		
Foreign Keys	AGENTID		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.REAGENT_PHONE (AGENTID NUMBER(4) NOT NULL, PHONE NUMBER(10), PRIMARY KEY(AGENTID, PHONE), FOREIGN KEY(AGENTID) REFERENCES REAGENT(AGENTID) ON DELETE CASCADE); INSERT INTO REAGENT_PHONE VALUES ('0001', '6774559889'); INSERT INTO REAGENT_PHONE VALUES ('0002', '2235567000');</pre>		

Count of records in the table	2
--------------------------------------	---

Name of the table	REAGENT_MAINCONTACT		
Description	This table provides the names of personal contacts that Suedan is familiar with within each real estate company.		
Attribute	Description	Type	<i>Examples of values</i>
AGENTID	Identifier of a real estate agent company	Number	Between 0 and 9999
MAIN_CONTACT	Personal contacts within the company that Suedan is familiar with	String	Bluff Lane
Primary Key	(AGENTID, MAIN_CONTACT)		
Foreign Keys	AGENTID		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.REAGENT_MAINCONTACT (AGENTID NUMBER(4) NOT NULL, MAIN_CONTACT VARCHAR2(50), PRIMARY KEY(AGENTID, MAIN_CONTACT), FOREIGN KEY(AGENTID) REFERENCES REAGENT(AGENTID) ON DELETE CASCADE); INSERT INTO REAGENT_MAINCONTACT VALUES ('0001', 'Ricardo Taylor'); INSERT INTO REAGENT_MAINCONTACT VALUES ('0001', 'Becky Clarke'); INSERT INTO REAGENT_MAINCONTACT VALUES ('0002', 'Timo Steffens');</pre>		
Count of records in the table	3		

Name of the table	DEVELOPED		
Description	This table describes information about properties that Suedan purchases and develops. Each purchase develops into a project.		
Attribute	Description	Type	<i>Examples of values</i>
PROJECTID	Identifier of a project	Number	Between 0 and 9999
REID	Identifier of a real estate property	Number	Between 0 and 9999

BUY_PRICE	Price that Suedan bought the property for	Number	Between 0 and 999999999
BUY_DATE	Date that Suedan bought the property	Date	2014-06-01
DATE_COMPLETED	Date that Suedan finished development	Date	2015-06-01
DEVELOP_COSTS	Total cost of development of the project	Number	Between 0 and 999999999
CURRENT_VALUE	Current market value of the developed property	Number	Between 0 and 999999999
Primary Key	PROJECTID		
Foreign Keys	REID		
Unique Keys	REID		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.DEVELOPED (PROJECTID NUMBER(4) NOT NULL, REID NUMBER(4) NOT NULL, BUY_PRICE NUMBER(9,2), BUY_DATE DATE, DATE_COMPLETED DATE, DEVELOP_COSTS NUMBER(9,2), CURRENT_VALUE NUMBER(9,2), PRIMARY KEY(PROJECTID), FOREIGN KEY(REID) REFERENCES UNDEVELOPED(REID) ON DELETE SET NULL, UNIQUE(REID)); INSERT INTO DEVELOPED VALUES('8700', '1001', '855000', date '2015-06-01', date '2016-07-30', '200000', '1300000'); INSERT INTO DEVELOPED VALUES('9900', '1003', '400000', date '2016-12-04', ", ", ");</pre>		
Count of records in the table	2		

Name of the table	STATUS		
Description	This table describes the current statuses that Developed properties can be.		
Attribute	Description	Type	Examples of values
STATUSID	Status ID	Number	0, 1, 2, 3, 4
PROJECTID	Identifier of a project	Number	Between 0 and 9999

STATUS	Status of property	String	In Development, Own, Rent, For Sale, Sold
Primary Key	STATUSID, PROJECTID		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.STATUS (STATUSID NUMBER(2) NOT NULL, PROJECTID NUMBER(4) NOT NULL, STATUS VARCHAR2(30) NOT NULL, PRIMARY KEY(STATUSID, PROJECTID)); INSERT INTO STATUS VALUES('1', '9900', 'OWN'); INSERT INTO STATUS VALUES('4', '8700', 'SOLD');</pre>		
Count of records in the table	4		

Name of the table	ARCHITECT		
Description	This table provides contact information about the architectural companies that Suedan does business with.		
Attribute	Description	Type	<i>Examples of values</i>
ARCHID	Identifier of an architectural company	Number	Between 0 and 9999
COMPANY_NAME	Name of company	String	Coney Architects
ZIP	Address	Number	29929
STREET	Address	String	90 Needle Ln
Primary Key	ARCHID		
Foreign Keys	PROJECTID, ZIP		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.ARCHITECT (ARCHID NUMBER(4) NOT NULL, COMPANY_NAME VARCHAR2(60) NOT NULL, ZIP NUMBER(5), STREET VARCHAR2(30), PRIMARY KEY(ARCHID), FOREIGN KEY(ZIP) REFERENCES ZIP_CODE(ZIP)); INSERT INTO ARCHITECT VALUES('0010', 'REGINA ARCHITECTS', '02115', '80 GREENWAY ST'); INSERT INTO ARCHITECT VALUES('0020', 'BACHELOR ARCHITECTS', '19355', '60 LAUREL CIRCLE');</pre>		
Count of records in the table	2		

Name of the table	ARCHITECT_PHONE		
Description	This table provides the multiple phone numbers of each architectural company.		
Attribute	Description	Type	<i>Examples of values</i>
ARCHID	Identifier of an architectural company	Number	Between 0 and 9999
PHONE	Company's phone number(s)	Number	212-666-0000
Primary Key	(ARCHID, PHONE)		
Foreign Keys	ARCHID		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.ARCHITECT_PHONE (ARCHID NUMBER(4) NOT NULL, PHONE NUMBER(10), PRIMARY KEY(ARCHID, PHONE), FOREIGN KEY(ARCHID) REFERENCES ARCHITECT(ARCHID) ON DELETE CASCADE); INSERT INTO ARCHITECT_PHONE VALUES('0010', '6179990000'); INSERT INTO ARCHITECT_PHONE VALUES('0010', '6179990500'); INSERT INTO ARCHITECT_PHONE VALUES('0020', '6106402621'); INSERT INTO ARCHITECT_PHONE VALUES('0020', '4843566600');</pre>		
Count of records in the table	4		

Name of the table	ARCHITECT_MAINCONTACT		
Description	This table provides the names of personal contacts that Suedan is familiar with within each real estate company.		
Attribute	Description	Type	<i>Examples of values</i>
ARCHID	Identifier of an architectural company	Number	Between 0 and 9999
MAIN_CONTACT	Contact's name	String	Ronald Mu
Primary Key	(ARCHID, MAIN_CONTACT)		
Foreign Keys	ARCHID		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.ARCHITECT_MAINCONTACT (ARCHID NUMBER(4) NOT NULL, MAIN_CONTACT VARCHAR2(30), PRIMARY KEY(ARCHID, MAIN_CONTACT),</pre>		

	FOREIGN KEY(ARCHID) REFERENCES ARCHITECT(ARCHID) ON DELETE CASCADE); INSERT INTO ARCHITECT_MAINCONTACT VALUES('0010', 'REGINA XI'); INSERT INTO ARCHITECT_MAINCONTACT VALUES('0010', 'RICHARD SIMMONS'); INSERT INTO ARCHITECT_MAINCONTACT VALUES('0020', 'JESSICA ANNE');
Count of records in the table	3

<i>Name of the table</i>	CONSTRUCTION		
Description	This table provides contact information about construction companies that Suedan does business with.		
Attribute	Description	Type	<i>Examples of values</i>
CONSTRID	Identifier of a construction company	Number	Between 0 and 9999
COMPANY_NAME	Name of company	String	Neptun Construction
ZIP	Address	Number	29929
STREET	Address	String	90 Needle Ln
Primary Key	CONSTRID		
Foreign Keys	PROJECTID, ZIP		
SQL Code	CREATE TABLE USERINSC521SU17_KZH75.CONSTRUCTION (CONSTRID NUMBER(4) NOT NULL, COMPANY_NAME VARCHAR2(30) NOT NULL, ZIP NUMBER(5), STREET VARCHAR2(50), PRIMARY KEY(CONSTRID), FOREIGN KEY(ZIP) REFERENCES ZIP_CODE(ZIP)); INSERT INTO CONSTRUCTION VALUES('0100', 'BAYPORT CONSTRUCTION CO', '02115', '56 HAIDIAN ROAD'); INSERT INTO CONSTRUCTION VALUES('0200', 'ZAMBONI CONSTRUCTION CO', '19355', '9000 LANCASTER AVE');		
Count of records in the table	2		

Name of the table	CONSTRUCTION_PHONE		
Description	This table provides the multiple phone numbers of each construction company.		
Attribute	Description	Type	<i>Examples of values</i>
CONSTRID	Identifier of a construction company	Number	Between 0 and 9999
PHONE	Company's phone number(s)	Number	212-666-0000
Primary Key	(CONSTRID, PHONE)		
Foreign Keys	CONSTRID		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.CONSTRUCTION_PHONE (CONSTRID NUMBER(4) NOT NULL, PHONE NUMBER(10), PRIMARY KEY(CONSTRID, PHONE), FOREIGN KEY(CONSTRID) REFERENCES CONSTRUCTION(CONSTRID) ON DELETE CASCADE); INSERT INTO CONSTRUCTION_PHONE VALUES('0100', '1222333890'); INSERT INTO CONSTRUCTION_PHONE VALUES('0100', '5677889922'); INSERT INTO CONSTRUCTION_PHONE VALUES('0200', '6688990000');</pre>		
Count of records in the table	3		

Name of the table	CONSTRUCTION_MAINCONTACT		
Description	This table provides the names of personal contacts that Suedan is familiar with within each real estate company.		
Attribute	Description	Type	<i>Examples of values</i>
CONSTRID	Identifier of a construction company	Number	Between 0 and 9999
MAIN_CONTACT	Contact's name	String	Ronald Mu
Primary Key	(CONSTRID, MAIN_CONTACT)		
Foreign Keys	CONSTRID		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.CONSTRUCTION_MAINCONTACT (CONSTRID NUMBER(4) NOT NULL, MAIN_CONTACT VARCHAR2(30), PRIMARY KEY(CONSTRID, MAIN_CONTACT), FOREIGN KEY(CONSTRID) REFERENCES CONSTRUCTION(CONSTRID) ON DELETE CASCADE);</pre>		

	INSERT INTO CONSTRUCTION_MAINCONTACT VALUES('0100', 'LONA MALLEY'); INSERT INTO CONSTRUCTION_MAINCONTACT VALUES('0200', 'LUPE MARIE'); INSERT INTO CONSTRUCTION_MAINCONTACT VALUES('0200', 'ZIPPER TILLY');
Count of records in the table	3

Name of the table	DESIGNS		
Description	This table provides information about which architectural company and which construction company designed and built every Developed property.		
Attribute	Description	Type	Examples of values
ARCHID	Identifier of an architectural company	Number	Between 0 and 9999
CONSTRID	Identifier of a construction company	Number	Between 0 and 9999
PROJECTID	Identifier of a project	Number	Between 0 and 9999
Primary Key	(ARCHID, CONSTRID, PROJECTID)		
Foreign Keys	ARCHID, CONSTRID, PROJECTID		
SQL Code	CREATE TABLE USERINSC521SU17_KZH75.DESIGNS (ARCHID NUMBER(4) NOT NULL, CONSTRID NUMBER(4) NOT NULL, PROJECTID NUMBER(4) NOT NULL, PRIMARY KEY(ARCHID, CONSTRID, PROJECTID), FOREIGN KEY(ARCHID) REFERENCES ARCHITECT(ARCHID), FOREIGN KEY(CONSTRID) REFERENCES CONSTRUCTION(CONSTRID), FOREIGN KEY(PROJECTID) REFERENCES DEVELOPED(PROJECTID)); INSERT INTO DESIGNS VALUES ('0010', '0100', '8700'); INSERT INTO DESIGNS VALUES ('0020', '0200', '9900');		
Count of records in the table	2		

Name of the table	EMPLOYEE		
Description	This table provides information about the employees that work for Suedan.		
Attribute	Description	Type	<i>Examples of values</i>
EMPLOYID	Identifier of individual employees	Number	Between 0 and 9999
FNAME	First name of employee	String	Mari
INITIAL	Middle initial of employee	String	B
LNAME	Last name of employee	String	Bales
DOB	Date of birth of employee	Date	1990-02-02
ZIP	Address	Number	02115
STREET	Address	String	8 Haidian Ave
M_EMPLOYID	Employee ID of the employee's manager	Number	1111
Primary Key	EMPLOYID		
Foreign Keys	EMPLOYID, ZIP		
SQL Code	<pre> CREATE TABLE USERINSC521SU17_KZH75.EMPLOYEE (EMPLOYID NUMBER(4) NOT NULL, FNAME VARCHAR2(15), INITIAL VARCHAR2(1), LNAME VARCHAR2(25), DOB VARCHAR2(10), ZIP NUMBER(5), STREET VARCHAR2(50), M_EMPLOYID NUMBER(4) DEFAULT('1111'), PRIMARY KEY(EMPLOYID), FOREIGN KEY(EMPLOYID) REFERENCES EMPLOYEE(EMPLOYID), FOREIGN KEY(ZIP) REFERENCES ZIP_CODE(ZIP)); INSERT INTO EMPLOYEE VALUES('1112', 'MARI', 'B', 'BALES', '86-06-23', '11705', '34 MAIN STREET', '1111'); INSERT INTO EMPLOYEE VALUES('1113', 'KIMBERLY', 'A', 'SIEGEL', '90-09-09', '29910', '3 KENNET DRIVE', '1114'); </pre>		
Count of records in the table	2		

Name of the table	EMPLOYEE_PHONE		
Description	This table provides the multiple phone numbers of each employee of Suedan.		
Attribute	Description	Type	Examples of values
EMPLOYID	Identifier of individual employees	Number	Between 0 and 9999
PHONE	Employee's phone number(s)	Number	212-666-0000
Primary Key	(EMPLOYID, PHONE)		
Foreign Keys	EMPLOYID		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.EMPLOYEE_PHONE (EMPLOYID NUMBER(4) NOT NULL, PHONE NUMBER(10), PRIMARY KEY(EMPLOYID, PHONE), FOREIGN KEY(EMPLOYID) REFERENCES EMPLOYEE(EMPLOYID)); INSERT INTO EMPLOYEE_PHONE VALUES('1112', '4843565512'); INSERT INTO EMPLOYEE_PHONE VALUES('1113', '9087638888');</pre>		
Count of records in the table	2		

Name of the table	WORKS_ON		
Description	This table provides information about which employees work on which project.		
Attribute	Description	Type	Examples of values
PROJECTID	Identifier of a project	Number	Between 0 and 9999
EMPLOYID	Identifier of individual employees	Number	Between 0 and 9999
Primary Key	(PROJECTID, EMPLOYID)		
Foreign Keys	PROJECTID, EMPLOYID		
SQL Code	<pre>CREATE TABLE USERINSC521SU17_KZH75.WORKS_ON (PROJECTID NUMBER(4) NOT NULL, EMPLOYID NUMBER(4), PRIMARY KEY(PROJECTID, EMPLOYID), FOREIGN KEY(PROJECTID) REFERENCES DEVELOPED(PROJECTID), FOREIGN KEY(EMPLOYID) REFERENCES EMPLOYEE(EMPLOYID)); INSERT INTO WORKS_ON VALUES('8700', '1112');</pre>		

	INSERT INTO WORKS_ON VALUES('9900', '1113');
Count of records in the table	2

Name of the table	REPORTS TO		
Description	This table provides information about the role of manager and subordinate in the Employee table.		
Attribute	Description	Type	Examples of values
EMPLOYID	Identifier of individual employees	Number	Between 0 and 9999
M_EMPLOYID	Employee ID of the employee's manager	Number	1111
Primary Key	(EMPLOYID, M_EMPLOYID)		
Foreign Keys	EMPLOYID		
SQL Code	CREATE TABLE USERINSC521SU17_KZH75.REPORTS_TO (EMPLOYID NUMBER(4), M_EMPLOYID NUMBER(4), PRIMARY KEY(EMPLOYID, M_EMPLOYID), FOREIGN KEY(EMPLOYID) REFERENCES EMPLOYEE(EMPLOYID)); INSERT INTO REPORTS_TO VALUES('1112', '1111'); INSERT INTO REPORTS_TO VALUES('1113', '1114');		
Count of records in the table	2		

4. DATABASE NORMALIZATION

Assumptions and Constraints

- The following tables contain no nonprime attributes and satisfy the conditions for 1NF:
REAGENT_PHONE, REAGENT_MAINCONTACT, ARCHITECT_PHONE,
ARCHITECT_MAINCONTACT, CONSTRUCTION_PHONE,
CONSTRUCTION_MAINCONTACT, DESIGNS, EMPLOYEE_PHONE, WORKS_ON,
REPORTS_TO

Tables Normal Form

<i>Name of the table</i>	ZIP_CODE	
Functional Dependencies	zip → city; zip → state	
Primary Key	zip	zip uniquely identifies any other attribute in this table
	Decision	Reason
First Normal Form	Yes	All cells contain atomic values and no repeating groups
Second Normal Form	Yes	Table is in 1NF and every nonprime attribute is fully dependent on single attribute key
Third Normal Form	Yes	Table is in 2NF and there are no transitive dependencies
Boyce Codd Normal Form	Yes	Table is in 3NF and every determinate is a primary key

<i>Name of the table</i>	UNDEVELOPED	
Functional Dependencies	reid → zip; reid → street; reid → type; reid → physical_description; reid → lastsold_date; reid → lastsold_price; reid → asking_price; reid → property_taxes; (zip, asking_price) → property_taxes	
Primary Key	reid	reid uniquely identifies any other attribute in this table
	Decision	Reason
First Normal Form	Yes	All cells contain atomic values and no repeating groups
Second Normal Form	Yes	Table is in 1NF and every nonprime attribute is fully dependent on single attribute key

Third Normal Form	Yes	Table is in 2NF and there are no transitive dependencies
Boyce Codd Normal Form	No	Zip, asking_price is not a candidate key in the (zip, asking_price) → property_taxes dependency

<i>Name of the table</i>	BUILDING	
Functional Dependencies	reid → sqft; reid → stories; reid → year_built; reid → physical_description	
Primary Key	reid	reid uniquely identifies any other attribute in this table
	Decision	Reason
First Normal Form	Yes	All cells contain atomic values and no repeating groups
Second Normal Form	Yes	Table is in 1NF and every nonprime attribute is fully dependent on single attribute key
Third Normal Form	Yes	Table is in 2NF and there are no transitive dependencies
Boyce Codd Normal Form	Yes	Table is in 3NF and every determinate is a primary key

<i>Name of the table</i>	LOT	
Functional Dependencies	reid → acres; reid → physical_description	
Primary Key	reid	reid uniquely identifies any other attribute in this table
	Decision	Reason
First Normal Form	Yes	All cells contain atomic values and no repeating groups
Second Normal Form	Yes	Table is in 1NF and every nonprime attribute is fully dependent on single attribute key
Third Normal Form	Yes	Table is in 2NF and there are no transitive dependencies
Boyce Codd Normal Form	Yes	Table is in 3NF and every determinate is a primary key

<i>Name of the table</i>	REAGENT	
Functional Dependencies	agentid → reid; agentid → company_name; agentid → zip;	

	agentid → street	
Primary Key	agentid	agentid uniquely identifies any other attribute in this table
	Decision	Reason
First Normal Form	Yes	All cells contain atomic values and no repeating groups
Second Normal Form	Yes	Table is in 1NF and every nonprime attribute is fully dependent on single attribute key
Third Normal Form	Yes	Table is in 2NF and there are no transitive dependencies
Boyce Codd Normal Form	Yes	Table is in 3NF and every determinate is a primary key

<i>Name of the table</i>	DEVELOPED	
Functional Dependencies	projectid → reid; projectid → buy_price; projectid → buy_date; projectid → date_completed; projectid → develop_costs; projectid → current_value	
Primary Key	projectid	projectid uniquely identifies any other attribute in this table
	Decision	Reason
First Normal Form	Yes	All cells contain atomic values and no repeating groups
Second Normal Form	Yes	Table is in 1NF and every nonprime attribute is fully dependent on single attribute key
Third Normal Form	Yes	Table is in 2NF and there are no transitive dependencies
Boyce Codd Normal Form	Yes	Table is in 3NF and every determinate is a primary key

<i>Name of the table</i>	STATUS	
Functional Dependencies	statusid → reid; statusid → status	
Primary Key	statusid	statusid uniquely identifies any other attribute in this table
	Decision	Reason
First Normal Form	Yes	All cells contain atomic values and no repeating groups
Second Normal Form	Yes	Table is in 1NF and every nonprime attribute is fully dependent on single attribute key

Third Normal Form	Yes	Table is in 2NF and there are no transitive dependencies
Boyce Codd Normal Form	Yes	Table is in 3NF and every determinate is a primary key

<i>Name of the table</i>	ARCHITECT	
Functional Dependencies	archid → company_name archid → zip archid → street	
Primary Key	archid	archid uniquely identifies any other attribute in this table
	Decision	Reason
First Normal Form	Yes	All cells contain atomic values and no repeating groups
Second Normal Form	Yes	Table is in 1NF and every nonprime attribute is fully dependent on single attribute key
Third Normal Form	Yes	Table is in 2NF and there are no transitive dependencies
Boyce Codd Normal Form	Yes	Table is in 3NF and every determinate is a primary key

<i>Name of the table</i>	CONSTRUCTION	
Functional Dependencies	constrid → company_name; constrid → zip; constrid → street	
Primary Key	constrid	constrid uniquely identifies any other attribute in this table
	Decision	Reason
First Normal Form	Yes	All cells contain atomic values and no repeating groups
Second Normal Form	Yes	Table is in 1NF and every nonprime attribute is fully dependent on single attribute key
Third Normal Form	Yes	Table is in 2NF and there are no transitive dependencies
Boyce Codd Normal Form	Yes	Table is in 3NF and every determinate is a primary key

<i>Name of the table</i>	EMPLOYEE	
Functional Dependencies	employid → fname; employid → minitial employid → lname	

	employid → dob employid → zip employid → street employid → m_employid	
Primary Key	employid	employid uniquely identifies any other attribute in this table
	Decision	Reason
First Normal Form	Yes	All cells contain atomic values and no repeating groups
Second Normal Form	Yes	Table is in 1NF and every nonprime attribute is fully dependent on single attribute key
Third Normal Form	Yes	Table is in 2NF and there are no transitive dependencies
Boyce Codd Normal Form	Yes	Table is in 3NF and every determinate is a primary key

5. SQL QUERIES

Query 1			
English version	Find all undeveloped property addresses that are lots between 0.5 and 3 acres.		
SQL sentence	SELECT U.STREET, U.ZIP, Z.CITY, Z.STATE FROM UNDEVELOPED U, LOT L, ZIP_CODE Z WHERE U.REID=L.REID AND U.ZIP=Z.ZIP AND L.ACRES BETWEEN 0.5 AND 3;		
Example of returned rows (screen caption)	STREET	ZIP CITY	ST
	-----	-----	---
	9 9th Ave	22601 WINCHESTER	VA
	5 Vonnie Dr	29577 MYRTLE BEACH	SC
	99 Roark Circle	29464 MOUNT PLEASANT	SC
	3244 Una Circle	18360 STODDSBURG	PA
	73892 Irvin Circle	7712 ASBURY PARK	NJ
	3222 Roush Dr	8330 MAYS LANDING	NJ
	222 Bond Rawls Road	7076 SCOTCH PLAINS	NJ
	31 5th Ave	23228 HENRICO	VA
	90 Julie Road	28803 ASHVILLE	NC
	8 KING ROAD	19355 MALVERN	PA
	600 FRAZER LANE	11705 BAYPORT	NY
	11 rows selected.		
Relational Algebra Query	Π U.STREET, U.ZIP, Z.CITY, Z.STATE (σ 0.5 \geq L.ACRES \geq 3 (ρ_L (LOT) \bowtie L.REID=U.REID ρ_U (UNDEVELOPED) \bowtie U.ZIP=Z.ZIP ρ_U (ZIP_CODE)))		

Query 2													
English version	Find all employee names and phone numbers with manager ID of 1111.												
SQL sentence	SELECT E.FNAME, E.LNAME, EP.PHONE FROM EMPLOYEE E, EMPLOYEE_PHONE EP, REPORTS_TO R WHERE R.M_EMPLOYID=1111 AND R.EMPLOYID=E.EMPLOYID AND E.EMPLOYID=EP.EMPLOYID;												
Example of returned rows (screen caption)	<table><tr><th>FNAME</th><th>LNAME</th><th>PHONE</th></tr><tr><td>MARI</td><td>BALES</td><td>4843565512</td></tr><tr><td>VERONICA</td><td>BALES</td><td>1222339001</td></tr><tr><td>KENY</td><td>WILLIAMS</td><td>3498239822</td></tr></table>	FNAME	LNAME	PHONE	MARI	BALES	4843565512	VERONICA	BALES	1222339001	KENY	WILLIAMS	3498239822
FNAME	LNAME	PHONE											
MARI	BALES	4843565512											
VERONICA	BALES	1222339001											
KENY	WILLIAMS	3498239822											
Relational Algebra Query	Π E.FNAME, E.LNAME, EP.PHONE (σ R.M_EMPLOYID=1111 (ρ_R (REPORTS_TO) \bowtie R.EMPLOYID=E.EMPLOYID ρ_E (EMPLOYEE) \bowtie E.EMPLOYID=EP.EMPLOYID ρ_{EP} (EMPLOYEE_PHONE)))												

Query 3																																					
English version	Find all undeveloped properties' ID number, type (lot or building), and city located in the state of New Jersey.																																				
SQL sentence	SELECT U.REID, U.TYPE, Z.CITY FROM UNDEVELOPED U, ZIP_CODE Z WHERE U.ZIP=Z.ZIP AND Z.STATE='NJ';																																				
Example of returned rows (screen caption)	<table><tr><th></th><th>REID</th><th>TYPE</th><th>CITY</th></tr><tr><td>1</td><td>1012</td><td>1</td><td>ASBURY PARK</td></tr><tr><td>2</td><td>1013</td><td>1</td><td>MAYS LANDING</td></tr><tr><td>3</td><td>1014</td><td>1</td><td>SCOTCH PLAINS</td></tr><tr><td>4</td><td>1015</td><td>1</td><td>FORT LEE</td></tr><tr><td>5</td><td>1036</td><td>2</td><td>ASBURY PARK</td></tr><tr><td>6</td><td>1037</td><td>2</td><td>MAYS LANDING</td></tr><tr><td>7</td><td>1038</td><td>2</td><td>SCOTCH PLAINS</td></tr><tr><td>8</td><td>1039</td><td>2</td><td>FORT LEE</td></tr></table>		REID	TYPE	CITY	1	1012	1	ASBURY PARK	2	1013	1	MAYS LANDING	3	1014	1	SCOTCH PLAINS	4	1015	1	FORT LEE	5	1036	2	ASBURY PARK	6	1037	2	MAYS LANDING	7	1038	2	SCOTCH PLAINS	8	1039	2	FORT LEE
	REID	TYPE	CITY																																		
1	1012	1	ASBURY PARK																																		
2	1013	1	MAYS LANDING																																		
3	1014	1	SCOTCH PLAINS																																		
4	1015	1	FORT LEE																																		
5	1036	2	ASBURY PARK																																		
6	1037	2	MAYS LANDING																																		
7	1038	2	SCOTCH PLAINS																																		
8	1039	2	FORT LEE																																		
Relational Algebra Query	Π U.REID, U.TYPE, Z.CITY (σ Z.STATE='NJ' (ρ_U (UNDEVELOPED) \bowtie U.ZIP=Z.ZIP ρ_Z (ZIP_CODE))))																																				

<i>Query 4</i>	
English version	Find the maximum amount of money Suedan spent on Development costs.
SQL sentence	<pre>SELECT MAX(DEVELOPMENT_COSTS) FROM DEVELOPED;</pre>
Example of returned rows (screen caption)	<pre>MAX(DEVELOPMENT_COSTS) ----- 500000</pre>
Relational Algebra Query	$F_{\max(\text{development_costs})}(\text{Developed})$

<i>Query 5</i>	
English version	Find the count of properties that Suedan spent between 100,000 and 300,000 on development costs grouped by development costs.
SQL sentence	<pre>SELECT DEVELOPMENT_COSTS, COUNT(DEVELOPMENT_COSTS) FROM DEVELOPED GROUP BY DEVELOPMENT_COSTS HAVING DEVELOPMENT_COSTS BETWEEN 100000 AND 300000;</pre>

Example of returned rows (screen caption)	\bowtie DEVELOPMENT_COSTS	\bowtie COUNT(DEVELOPMENT_COSTS)
	1	300000
	2	100000
	3	200000
	4	230000
	5	120000
	6	107000
Relational Algebra Query	$\gamma_{\text{development_costs}} \text{ F}_{\text{count(development_costs)}} \Pi_{\text{development_costs}} (\sigma_{\text{DEVELOPMENT_COSTS} \geq 100000 \wedge \text{DEVELOPMENT_COSTS} \leq 300000} (\text{DEVELOPED}))$	

Query 6											
English version	Find the youngest and oldest employee.										
SQL sentence	<pre>(SELECT FNAME, LNAME, DOB FROM EMPLOYEE WHERE DOB=(SELECT MAX(DOB) FROM EMPLOYEE)) UNION (SELECT FNAME, LNAME, DOB FROM EMPLOYEE WHERE DOB=(SELECT MIN(DOB) FROM EMPLOYEE));</pre>										
Example of returned rows (screen caption)	<table> <thead> <tr> <th>FNAME</th><th>LNAME</th><th>DOB</th></tr> </thead> <tbody> <tr> <td>CLINT</td><td>MASON</td><td>49-02-19</td></tr> <tr> <td>JOHANNA</td><td>DAVIS</td><td>95-01-01</td></tr> </tbody> </table>		FNAME	LNAME	DOB	CLINT	MASON	49-02-19	JOHANNA	DAVIS	95-01-01
FNAME	LNAME	DOB									
CLINT	MASON	49-02-19									
JOHANNA	DAVIS	95-01-01									
Relational Algebra Query	$\Pi \text{ FNAME, LNAME, DOB } (\sigma_{\text{DOB}=\text{F MAX(DOB)}} (\text{Employee})) \cup \Pi \text{ FNAME, LNAME, DOB } (\sigma_{\text{DOB}=\text{F MIN(DOB)}} (\text{Employee}))$										

Query 7	
English version	Find the name of the construction and architect companies that developed the building at 9 White St.
SQL sentence	<pre>SELECT A.COMPANY_NAME, C.COMPANY_NAME FROM ARCHITECT A, CONSTRUCTION C, DESIGNS DE WHERE DE.ARCHID=A.ARCHID AND DE.CONSTRID=C.CONSTRID AND DE.PROJECTID=(SELECT D.PROJECTID FROM UNDEVELOPED U, DEVELOPED D WHERE U.STREET='9 White St' AND U.REID=D.REID);</pre>

Example of returned rows (screen caption)	<table> <tr> <td>COMPANY_NAME</td><td>COMPANY_NAME</td></tr> <tr> <td>SELINA ARCHITECTS</td><td>DORA CONSTRUCTION CO</td></tr> </table>	COMPANY_NAME	COMPANY_NAME	SELINA ARCHITECTS	DORA CONSTRUCTION CO
COMPANY_NAME	COMPANY_NAME				
SELINA ARCHITECTS	DORA CONSTRUCTION CO				
Relational Algebra Query	$\Pi A.COMPANY_NAME, C.COMPANY_NAME (\sigma \rho_A (ARCHITECT) \bowtie A.ARCHID=DE.ARCHID$ $\rho_{DE} (DESIGNS) \bowtie DE.CONSTRID=C.CONSTRID \rho_C$ $(CONSTRUCTION) \rho_{DE} (DESIGNS) \bowtie DE.PROJECTID=$ $(\Pi D.PROJECTID (\sigma U.STREET= '9 White St' (\rho_U (UNDEVELOPED)$ $\bowtie U.REID=D.REID \rho_D (DEVELOPED))))))$				

Query 8																																																																																																																																																										
English version	Find the development data of all the properties that Suedan currently owns.																																																																																																																																																									
SQL sentence	SELECT D.*, S.STATUS FROM STATUS S, DEVELOPED D WHERE S.PROJECTID=D.PROJECTID AND S.STATUSID BETWEEN 0 AND 3;																																																																																																																																																									
Example of returned rows (screen caption)	<table><tr><th></th><th>PROJECTID</th><th>REID</th><th>BUYING_PRICE</th><th>BUYING_DATE</th><th>DATE_COMPLETED</th><th>DEVELOPMENT_COSTS</th><th>CURRENT_VALUE</th><th>STATUS</th></tr><tr><td>1</td><td>9000</td><td>1014</td><td>890000</td><td>05-APR-13</td><td>(null)</td><td>(null)</td><td>(null)</td><td>IN DEVELOPMENT</td></tr><tr><td>2</td><td>9100</td><td>1015</td><td>938000</td><td>13-AUG-05</td><td>04-SEP-07</td><td>500000</td><td>1442000</td><td>FOR SALE</td></tr><tr><td>3</td><td>9200</td><td>1016</td><td>1000000</td><td>16-DEC-10</td><td>01-DEC-11</td><td>350000</td><td>1354000</td><td>OWN</td></tr><tr><td>4</td><td>9300</td><td>1017</td><td>450000</td><td>16-MAR-09</td><td>16-APR-16</td><td>200000</td><td>654000</td><td>RENT</td></tr><tr><td>5</td><td>9400</td><td>1018</td><td>600000</td><td>08-AUG-16</td><td>(null)</td><td>(null)</td><td>(null)</td><td>IN DEVELOPMENT</td></tr><tr><td>6</td><td>9500</td><td>1019</td><td>304000</td><td>16-JUL-00</td><td>16-JUL-02</td><td>120000</td><td>427000</td><td>OWN</td></tr><tr><td>7</td><td>9600</td><td>1020</td><td>500400</td><td>20-JUN-07</td><td>(null)</td><td>(null)</td><td>(null)</td><td>IN DEVELOPMENT</td></tr><tr><td>8</td><td>6000</td><td>1023</td><td>450400</td><td>14-APR-08</td><td>14-APR-12</td><td>400000</td><td>854400</td><td>FOR SALE</td></tr><tr><td>9</td><td>6100</td><td>1024</td><td>1009000</td><td>24-AUG-11</td><td>24-AUG-15</td><td>300000</td><td>1310000</td><td>FOR SALE</td></tr><tr><td>10</td><td>6200</td><td>1025</td><td>3456000</td><td>13-AUG-12</td><td>(null)</td><td>(null)</td><td>(null)</td><td>IN DEVELOPMENT</td></tr><tr><td>11</td><td>6400</td><td>1027</td><td>5500000</td><td>27-MAY-14</td><td>(null)</td><td>300000</td><td>5804000</td><td>OWN</td></tr><tr><td>12</td><td>6600</td><td>1029</td><td>500000</td><td>29-JUN-01</td><td>29-JUN-04</td><td>100000</td><td>604000</td><td>FOR SALE</td></tr><tr><td>13</td><td>6700</td><td>1030</td><td>789000</td><td>23-JUL-04</td><td>23-JUL-06</td><td>107000</td><td>900000</td><td>FOR SALE</td></tr><tr><td>14</td><td>6800</td><td>1031</td><td>3833000</td><td>30-JAN-11</td><td>(null)</td><td>(null)</td><td>(null)</td><td>IN DEVELOPMENT</td></tr><tr><td>15</td><td>6900</td><td>1032</td><td>200090</td><td>30-APR-03</td><td>(null)</td><td>(null)</td><td>(null)</td><td>IN DEVELOPMENT</td></tr><tr><td>16</td><td>9900</td><td>1003</td><td>400000</td><td>04-DEC-16</td><td>(null)</td><td>(null)</td><td>(null)</td><td>OWN</td></tr></table>		PROJECTID	REID	BUYING_PRICE	BUYING_DATE	DATE_COMPLETED	DEVELOPMENT_COSTS	CURRENT_VALUE	STATUS	1	9000	1014	890000	05-APR-13	(null)	(null)	(null)	IN DEVELOPMENT	2	9100	1015	938000	13-AUG-05	04-SEP-07	500000	1442000	FOR SALE	3	9200	1016	1000000	16-DEC-10	01-DEC-11	350000	1354000	OWN	4	9300	1017	450000	16-MAR-09	16-APR-16	200000	654000	RENT	5	9400	1018	600000	08-AUG-16	(null)	(null)	(null)	IN DEVELOPMENT	6	9500	1019	304000	16-JUL-00	16-JUL-02	120000	427000	OWN	7	9600	1020	500400	20-JUN-07	(null)	(null)	(null)	IN DEVELOPMENT	8	6000	1023	450400	14-APR-08	14-APR-12	400000	854400	FOR SALE	9	6100	1024	1009000	24-AUG-11	24-AUG-15	300000	1310000	FOR SALE	10	6200	1025	3456000	13-AUG-12	(null)	(null)	(null)	IN DEVELOPMENT	11	6400	1027	5500000	27-MAY-14	(null)	300000	5804000	OWN	12	6600	1029	500000	29-JUN-01	29-JUN-04	100000	604000	FOR SALE	13	6700	1030	789000	23-JUL-04	23-JUL-06	107000	900000	FOR SALE	14	6800	1031	3833000	30-JAN-11	(null)	(null)	(null)	IN DEVELOPMENT	15	6900	1032	200090	30-APR-03	(null)	(null)	(null)	IN DEVELOPMENT	16	9900	1003	400000	04-DEC-16	(null)	(null)	(null)	OWN
	PROJECTID	REID	BUYING_PRICE	BUYING_DATE	DATE_COMPLETED	DEVELOPMENT_COSTS	CURRENT_VALUE	STATUS																																																																																																																																																		
1	9000	1014	890000	05-APR-13	(null)	(null)	(null)	IN DEVELOPMENT																																																																																																																																																		
2	9100	1015	938000	13-AUG-05	04-SEP-07	500000	1442000	FOR SALE																																																																																																																																																		
3	9200	1016	1000000	16-DEC-10	01-DEC-11	350000	1354000	OWN																																																																																																																																																		
4	9300	1017	450000	16-MAR-09	16-APR-16	200000	654000	RENT																																																																																																																																																		
5	9400	1018	600000	08-AUG-16	(null)	(null)	(null)	IN DEVELOPMENT																																																																																																																																																		
6	9500	1019	304000	16-JUL-00	16-JUL-02	120000	427000	OWN																																																																																																																																																		
7	9600	1020	500400	20-JUN-07	(null)	(null)	(null)	IN DEVELOPMENT																																																																																																																																																		
8	6000	1023	450400	14-APR-08	14-APR-12	400000	854400	FOR SALE																																																																																																																																																		
9	6100	1024	1009000	24-AUG-11	24-AUG-15	300000	1310000	FOR SALE																																																																																																																																																		
10	6200	1025	3456000	13-AUG-12	(null)	(null)	(null)	IN DEVELOPMENT																																																																																																																																																		
11	6400	1027	5500000	27-MAY-14	(null)	300000	5804000	OWN																																																																																																																																																		
12	6600	1029	500000	29-JUN-01	29-JUN-04	100000	604000	FOR SALE																																																																																																																																																		
13	6700	1030	789000	23-JUL-04	23-JUL-06	107000	900000	FOR SALE																																																																																																																																																		
14	6800	1031	3833000	30-JAN-11	(null)	(null)	(null)	IN DEVELOPMENT																																																																																																																																																		
15	6900	1032	200090	30-APR-03	(null)	(null)	(null)	IN DEVELOPMENT																																																																																																																																																		
16	9900	1003	400000	04-DEC-16	(null)	(null)	(null)	OWN																																																																																																																																																		
Relational Algebra Query	$\Pi D*, S.STATUS (\sigma 0 \geq S.STATUSID \leq 3 (\rho_S (STATUS)$ $\bowtie S.PROJECTID=D.PROJECTID \rho_S (DEVELOPED))))$																																																																																																																																																									