Project for Database Design

Phase II. Relational Schema

Indhumadhi Suryanaryanan Suneesha Kudipudi
ixs150630@utdallas.edu sxk157430@utdallas.edu

Sindhujaa Ranganathan Chandra Babu
sxr152530@utdallas.edu

(Week 6-10: Sep 28-Oct.26)

0. Pre-Illumination

For clearly describing the relational schema design, we separate this report into four sections. In Section 1 we modify the original EER diagram and explain what are changed, respect to our Phase I EER diagram. And then, in Section 2 we give the relational schemas converted from our Phase I EER diagram with detailed mapping step by step. Section 3 is the documentation of relational schemas. This documentation mainly describes data type and format for each attribute in each relational schema. We also explain our assumptions for the documentation in this section. Finally, a short summary is given at the end of this report.

1. Modified EER diagram

Below are the modifications done:

- (0,N), a customer may sign contracts with multiple subdivisions.
- A floorplan may be used in several contracts.
- An existing customer may sign multiple contracts with a subdivision.
- A floorplan may be used for several inventory homes.

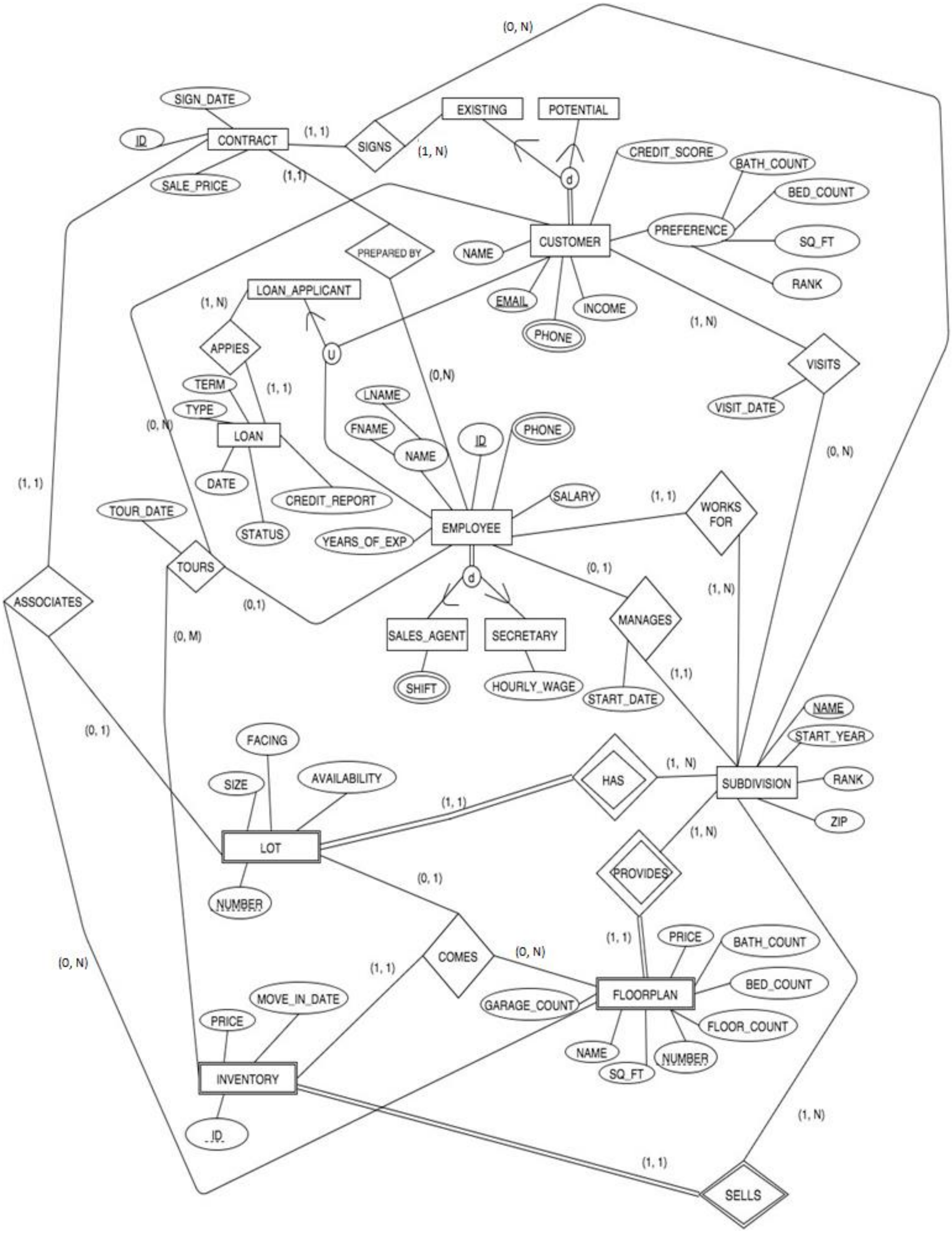


Figure 1. EER Design for Home Builder Companvl Database

2. Mapping Relational Schemas

We use seven-step algorithm to convert the basic EER model constructs into relations. The following are detailed mapping process.

2.1 Mapping of Regular Entity Types, Specializations.

REGULAR ENTITIES:

CONTRACT

ID	SIGN_DATE	SALE_PRICE

CUSTOMER

EMAIL	NAME	INCOME	CREDIT_SCORE	BATH_COUNT	BED_COUNT	SQ_FT	RANK

EMPLOYEE

ID	FNAME	LNAME	SALARY	YEARS_OF_EXP

LOAN

ID	TYPE	TERM	STATUS	DATE	CREDIT_REPORT

SUBDIVISION

NAME	START_YEAR	RANK	ZIP

SPECIALIZATION AND UNION:

1. Specialization between CUSTOMER, EXISTING and POTENTIAL

CUSTOMER

EMAIL	NAME	INCOME	CREDIT_SCORE	BATH_COUNT	BED_COUNT	SQ_FT	RANK	TYPE

EXISTING

EMAIL

POTENTIAL

EMAIL

We included a separate table for the subclasses with the primary key EMAIL of the superclass CUSTOMER as both the primary key and foreign key in the subclasses of POTENTIAL and EXISTING.

2. Specialization between EMPLOYEE, SALES_AGENT and SECRETARY

EMPLOYEE

ID	FNAME	LNAME	SALARY	E_TYPE	YEARS_OF_EXP

SALESAGENT

EMPLOYEE_ID	SHIFT

SECRETARY

EMP_ID	HOURLY_WAGE

We included a separate table for the subclasses with the primary key ID of the superclass EMPLOYEE as both the primary key and foreign key EMP_ID in the subclasses SALES_AGENT and SECRETARY along with their local attributes.

3.Union between CUSTOMER, EMPLOYEE and LOAN_APPLICANT

LOANAPPLICANT

ID

CUSTOMER

EMAIL	NAME	INCOME	CREDIT_SCORE	BATH_COUNT	BED_COUNT	SQ_FT	RANK	TYPE

LOANAPPLICANT_ID

EMPLOYEE

ID	FNAME	LNAME	SALARY	TYPE	LOANAPPLICANT_ID	YEARS_OF_EXP	SUBDIVISION_NAME

We included a surrogate key called LOAN_APP_ID as the primary key for LOAN_APPLICANT and include it as foreign key in the superclasses.

2.2 Mapping of Weak Entity Type

1. FLOORPLAN

SUBDIVISION_NAME	NUMBER	PRICE	BATH_COUNT	BED_COUNT	FLOOR_COUNT	SQ_FT	GARAGE_COUNT

2. LOT

SUBDIVISION_NAME	NUMBER	SIZE	FACING	AVAILABILITY

3. INVENTORY

SUBDIVISION_NAME	ID	PRICE	MOVE_IN_DATE

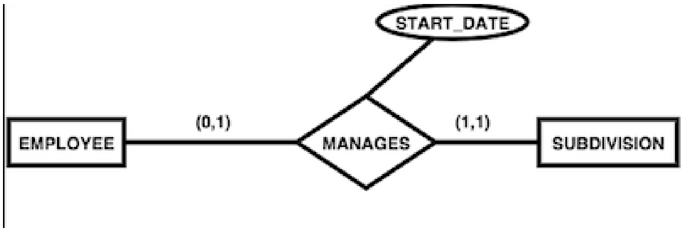
For mapping weak entities we include the primary key of the strong entity it belongs to as foreign key and along with the partial key as the new primary key.

2.3 Mapping of Binary 1:1 Relationship Types

The mapping method is exhibited in Table 1.

Table 1. Mapping Method to Binary 1:1 Relationship

Relation	Mapping Method

 START_DATE EMPLOYEE (0,1) MANAGES (1,1) SUBDIVISION	We use Foreign key approach to map this relationship. Since on SUNDIVISION side, the relationship is total participation, we include EMP_ID as a foreign key in SUBDIVISION, which is the primary key of Employees.

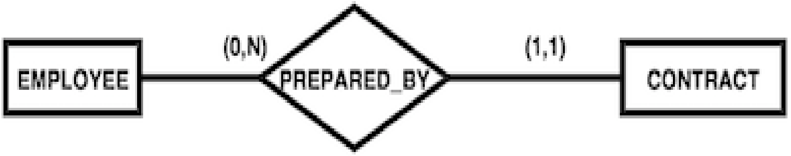
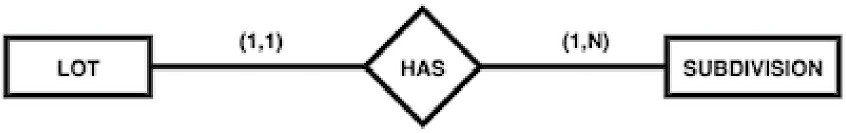
SUBDIVISION

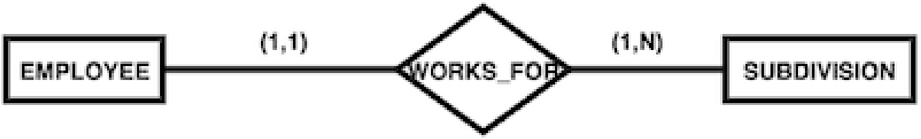
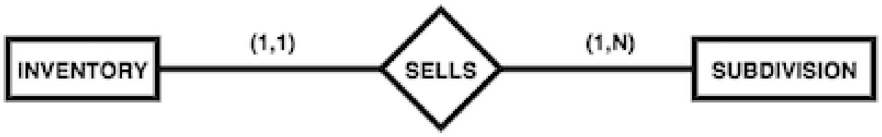
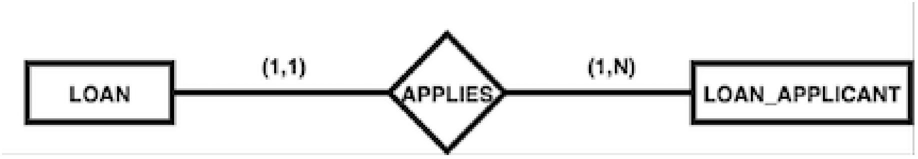
NAME	START_YEAR	RANK	ZIP	MANAGER_ID

2.4 Mapping of Binary 1:N Relationship Types

The mapping method is exhibited in Table 2.

Table 2. Mapping Method to Binary 1:N Relationship

Relation	**Mapping Method**
	The *N-side* of this relationship type is EMPLOYEE. Thus we include the primary key ID of the relation PREPARED_BY as foreign key in relation CONTRACT.
	The *N-side* of this relationship type is SUBDIVISION. Thus we include the primary key NAME of the relation HAS as foreign key in relation LOT.

	The *N-side* of this relationship type is SUBDIVISION. Thus we include the primary key NAME of the relation WORKS_FOR as foreign key in relation EMPLOYEE.
	The *N-side* of this relationship type is SUBDIVISION. Thus we include the primary key NAME of the relation SELLS as foreign key in relation INVENTORY.
	The *N-side* of this relationship type is LOAN_APPLICANT Thus we include the primary key LOAN_APP_ID of the relation APPLIES as foreign key in relation LOAN.

(example)

CONTRACT

ID	SIGN_DATE	SALE_PRICE	EMP_ID

LOT

SUB_NAME	NUMBER	SIZE	FACING	AVAILABILITY	SUB_NAME

EMPLOYEE

ID	FNAME	LNAME	SALARY	TYPE	LOAN_APP_ID	YEARS_OF_EXP	SUB_NAME

INVENTORY

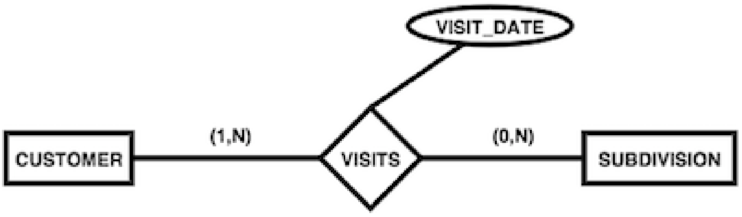
SUB_NAME	ID	PRICE	MOVE_IN_DATE

LOAN

ID	TYPE	TERM	STATUS	DATE	CREDIT_REPORT	LOAN_APP_ID

2.5 Mapping of Binary M:N Relationship Types

Table 1. Mapping Method to Binary 1:1 Relationship

Relation	Mapping Method
 VISIT_DATE CUSTOMER (1,N) VISITS (0,N) SUBDIVISION	We use a separate table for mapping the m:n relationship types with the primary keys of both the entities CUSTOMER and SUBDIVISION.

VISITS

CUST_EMAIL	SUB_NAME	VISIT_DATE

2.6 Mapping of Multi-valued Attributes

PHONE in CUSTOMER, PHONE in EMPLOYEE and SHIFT in SALES_AGENT are the multivalued attributes. For these, we create a separate table with the primary key of the entities as the foreign key and the multivalued attribute as the new combined primary key.

CUSTOMERPHONE

EMAIL	PHONE

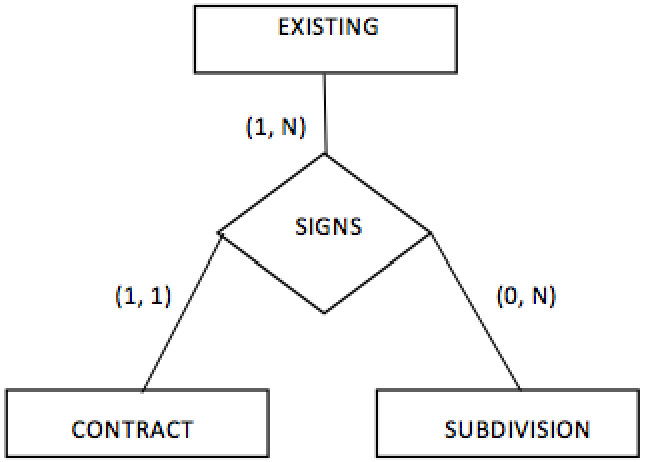
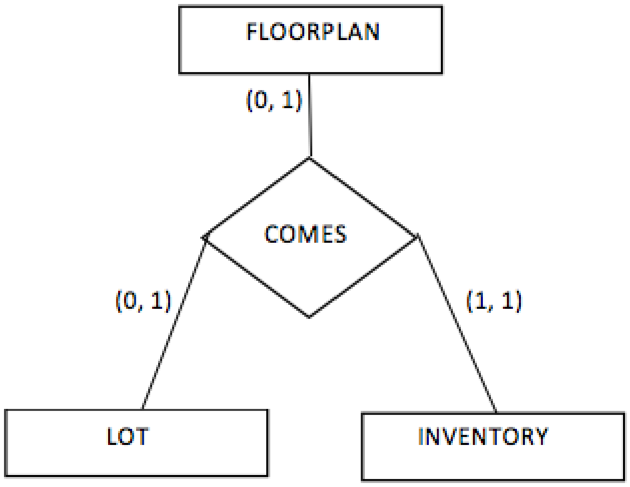
EMPLOYEEPHONE

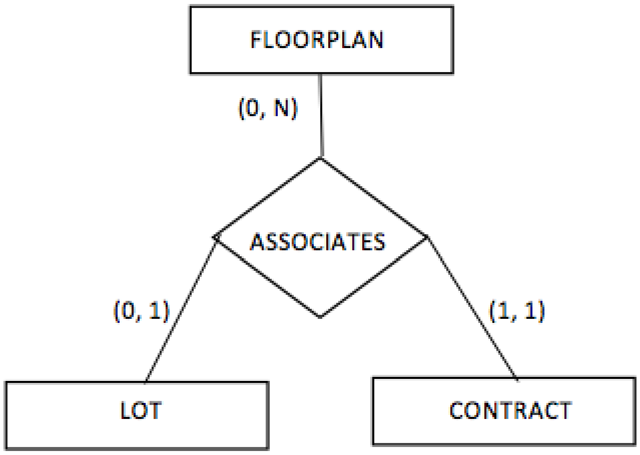
EMP_ID	PHONE

SALESAGENT

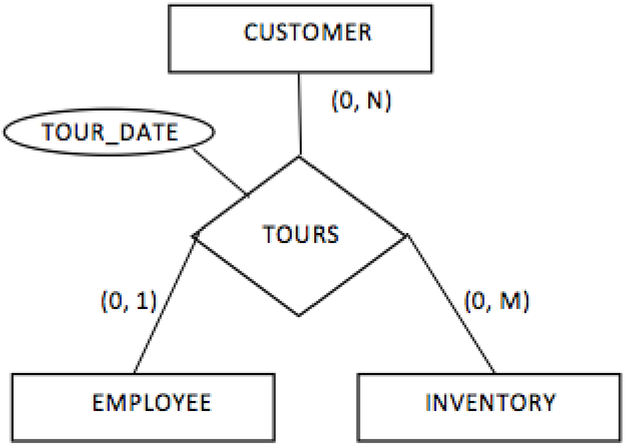
EMPLOYEE_ID	SHIFT

2.7 Mapping of N-ary Relationship Types

Relation	Mapping Method
 EXISTING (1, N) SIGNS (1, 1) (0, N) CONTRACT SUBDIVISION	We mapped the ternary relationship by creating a separate table with the keys of all the participating entities EMPLOYEE, CONTRACT and SUBDIVISION as the foreign keys and combining them as a single primary key.
 FLOORPLAN (0, 1) COMES (0, 1) (1, 1) LOT INVENTORY	We mapped the ternary relationship by creating a separate table with the keys of all the participating entities FLOORPLAN, LOT and INVENTORY as the foreign keys and combining them as a single primary key.

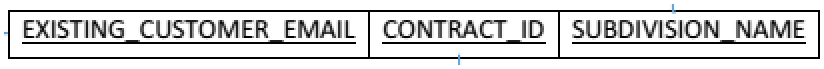


We mapped the ternary relationship by creating a separate table with the keys of all the participating entities FLOORPLAN, LOT and CONTRACT as the foreign keys and combining them as a single primary key.



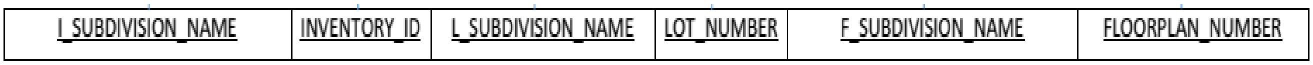
We mapped the ternary relationship by creating a separate table with the keys of all the participating entities CUSTOMER, EMPLOYEE and INVENTORY as the foreign keys and combining them as a single primary key.

SIGNS



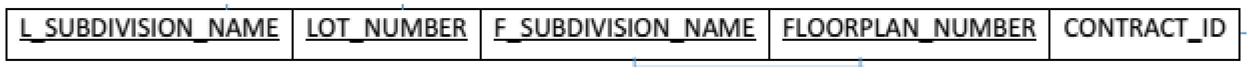
EXISTING_CUSTOMER_EMAIL	CONTRACT_ID	SUBDIVISION_NAME

COMES



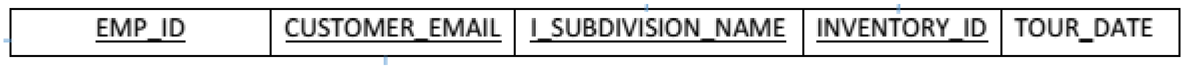
I_SUBDIVISION_NAME	INVENTORY_ID	L_SUBDIVISION_NAME	LOT_NUMBER	F_SUBDIVISION_NAME	FLOORPLAN_NUMBER

ASSOCIATES



L_SUBDIVISION_NAME	LOT_NUMBER	F_SUBDIVISION_NAME	FLOORPLAN_NUMBER	CONTRACT_ID

TOURS



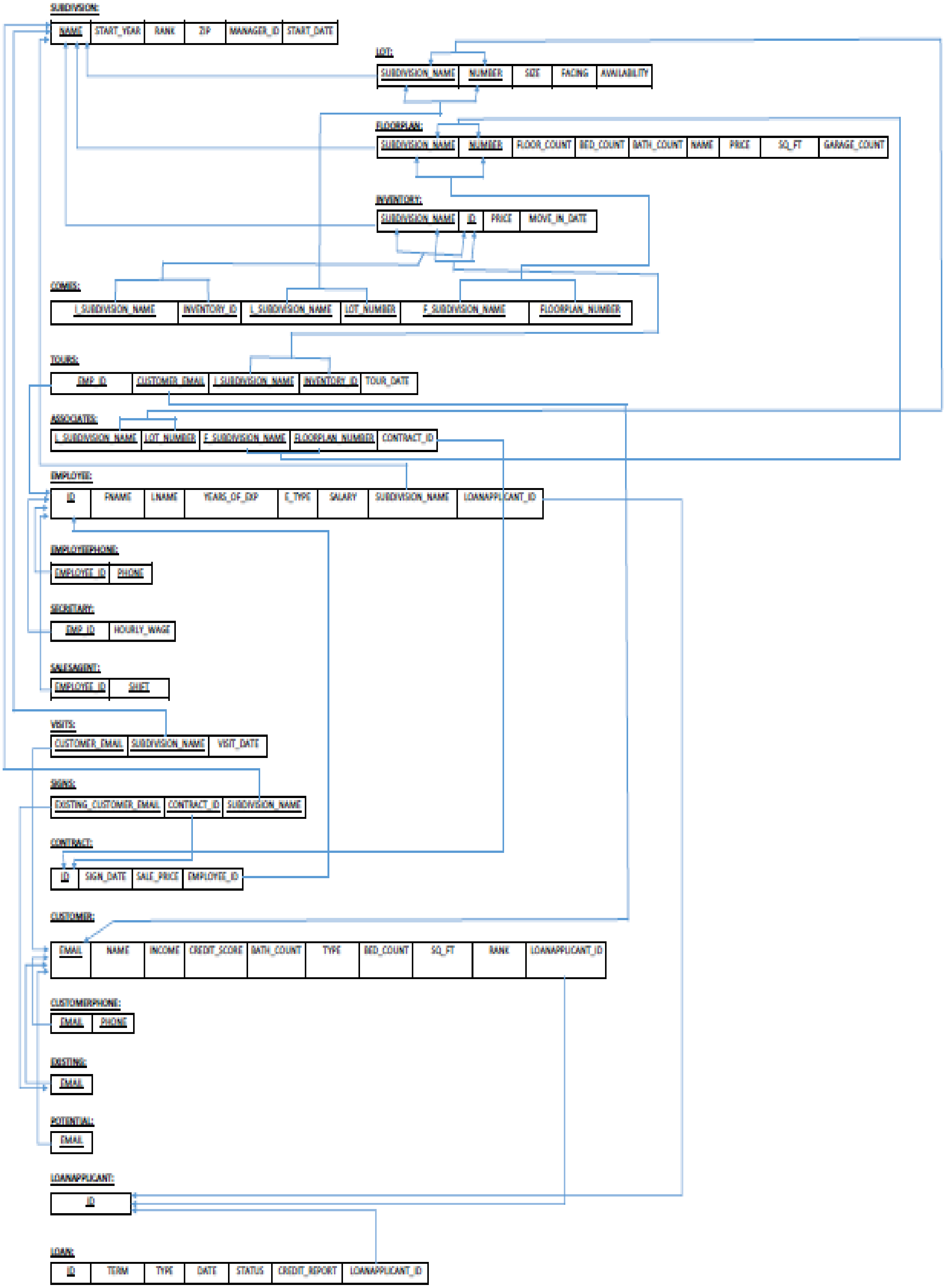
EMP_ID	CUSTOMER_EMAIL	I_SUBDIVISION_NAME	INVENTORY_ID	TOUR_DATE

2.8 Final Relation Schema of Home Builder Company Database

After seven steps mapping, we can get the final result of relation schema. Besides, we point out foreign keys by arrows from foreign key to the original keys between two relations.

Figure 5 displays all the relational schemas converted from Phase I EER diagram. **Clearer image is attached as a separate pdf to the project.**

Figure 5:



SUBDIVISION:

NAME	START_YEAR	RANK	ZIP	MANAGER_ID	START_DATE

LOT:

SUBDIVISION_NAME	NUMBER	SIZE	FACING	AVAILABILITY

FLOORPLAN:

SUBDIVISION_NAME	NUMBER	FLOOR_COUNT	BED_COUNT	BATH_COUNT	NAME	PRICE	SQ_FT	GARAGE_COUNT

INVENTORY:

SUBDIVISION_NAME	ID	PRICE	MOVE_IN_DATE

COMES:

I_SUBDIVISION_NAME	INVENTORY_ID	L_SUBDIVISION_NAME	LOT_NUMBER	F_SUBDIVISION_NAME	FLOORPLAN_NUMBER

TOURS:

EMP_ID	CUSTOMER_EMAIL	I_SUBDIVISION_NAME	INVENTORY_ID	TOUR_DATE

ASSOCIATES:

L_SUBDIVISION_NAME	LOT_NUMBER	F_SUBDIVISION_NAME	FLOORPLAN_NUMBER	CONTRACT_ID

EMPLOYEE:

ID	FNAME	LNAME	YEARS_OF_EXP	E_TYPE	SALARY	SUBDIVISION_NAME	LOANAPPLICANT_ID

EMPLOYEEPHONE:

EMPLOYEE_ID	PHONE

SECRETARY:

EMP_ID	HOURLY_WAGE

SALESAGENT:

EMPLOYEE_ID	SHIFT

VISITS:

CUSTOMER_EMAIL	SUBDIVISION_NAME	VISIT_DATE

SIGNS:

EXISTING_CUSTOMER_EMAIL	CONTRACT_ID	SUBDIVISION_NAME

CONTRACT:

ID	SIGN_DATE	SALE_PRICE	EMPLOYEE_ID

CUSTOMER:

EMAIL	NAME	INCOME	CREDIT_SCORE	BATH_COUNT	TYPE	BED_COUNT	SQ_FT	RANK	LOANAPPLICANT_ID

CUSTOMERPHONE:

EMAIL	PHONE

EXISTING:

EMAIL

POTENTIAL:

EMAIL

LOANAPPLICANT:

ID

LOAN:

ID	TERM	TYPE	DATE	STATUS	CREDIT_REPORT	LOANAPPLICANT_ID

3. Documentation for schemas

3.1 Explanation for format design

After mapping the EER diagram into relation schema that can be implemented in a relational DBMS like Oracle, we should also design the format of each attribute in every relation. Here we suppose that all the assumptions, explanations and limitations in phase I are also suitable for the design in this phase. Thus, we shall not repeat them. In this section, we only explain our assumptions for the data types and formats in the documentation. The rules are shown as follows:

- Data format for all EMPLOYEE IDs is an Integer type with exactly 12 digits i.e with Domain values (100000000000 to 999999999999).
- Data format for all SALARY is a Float type.
- Data format for all Employee Names (FNAME and LNAME) is a variable length String type with not more than 15 characters.
- Data format for all YEARS_OF_EXP is an Integer type with values varying from 0 to 55 only.
- Data format of E_TYPE is of String type with values either as ‘Sales agent’ or ‘Secretary’.
- Data format of SHIFT is a String type with exactly 3 characters having the Domain ('Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun').
- Data format of all SUBDIVISION.NAME is of Variable String with less than 25 characters.
- Data format of RANK is Integer type.
- Data format of all YEAR is 4 digit Integer.
- Data format of all ZIP is exactly 5 digit Integer.
- HOURLY_WAGE of SECRETARY is of Float type.
- Data Format of all PHONE is of String type with exactly 13 Chars of the form (XXX)XXX-XXXX.
- Data format of all the LOAN IDs is of Integer type with exactly 10 digits i.e with Domain values (1000000000 to 9999999999).
- Data format of LOAN TERM is a String of variable length less than 10 characters.
- Data format of LOAN TYPE is a String type of variable length less than 7 characters.
- Data format of all DATEs is a Date type of the form MM/DD/YYYY.
- Data format of LOAN.CREDIT_REPORT is of variable length String of not more than 10 characters.
- Data format of LOAN.STATUS is of variable length String of not more than 10 characters.
- Data format of all LOAN_APPLICANT.ID is of Integer type with 5 digits i.e with Domain values (10000 to 99999).
- Data format of LOT.NUMBER is of Integer type with exactly 4 digits i.e with Domain values (1000 to 9999).
- Data format of LOT.SIZE is of Integer type.

- Data format of AVAILABILITY is a single character with either Y or N as values, Y implying Yes, the LOT is available and N implying No, the LOT is not available.
- Data format of FACING is of String type with a variable length but not more than 5 characters with domain values as ('North','South','East','West').
- Data format of FLOORNAME.NUMBER should be of Integer type with exactly 3 digits i.e with Domain values (100 to 999).
- Data format of FLOOR_COUNT is of Integer type with domain values either 1,2 or 3.
- Data format of BED_COUNT should be of Integer type with domain values (1,2,3,4,5).
- Data format of BATH_COUNT is of Integer type with domain values (1,2,3,4,5).
- Data format of NAME should be of String of variable length no longer than 10 Characters.
- Data format of SQ_FT is of Integer with exactly 4 digits i.e with Domain values (1000 to 9999).
- Data format of GARAGE_COUNT should be of Integer type with domain values(1,2,3).
- Data format of all PRICE is of Integer type.
- Data format of INVENTORY.ID should be of Integer type with exactly 5 digits i.e with Domain values (10000 to 99999).
- Data format of all EMAIL is a variable String of the format 'varchar@varchar.com' of not more than 30 characters.

3.2 Format for Every Relation

Table 3 gives data type and format for each attribute in each relational schema.

Table 3. Format for Each Attribute

RELATION NAMES	ATTRIBUTES	DATATYPE
SUBDIVISION	NAME	String <=25 Chars
	START_YEAR	Integer = 4 Digits
	RANK	Integer
	ZIP	Integer = 5 Digits
	MANAGER_ID	12 digit Integer, Domain values(100000000000 to 999999999999)
	START_DATE	MM/DD/YYYY, Date type
CONTRACT	ID	Integer
	SIGN_DATE	MM/DD/YYYY, Date type
	SALE_PRICE	Integer
	EMPLOYEE_ID	12 digit Integer, Domain values(100000000000 to 999999999999)
EMPLOYEE	ID	12 digit Integer, Domain values(100000000000 to 999999999999)
	SALARY	Float
	FNAME	String <=15 Chars
	SUBDIVISION_NAME	String <=25 Chars

	LOANAPPLICANT_ID	Integer
	LNAME	String <=15 Chars
	YEARS_OF_EXP	Integer Domain(0 to 55)
	E_TYPE	String ,Domain('Sales agent','Secretary')
CUSTOMER	EMAIL	‘<variable chars>@<variable chars>.com’, String <= 30 chars
	NAME	String <= 50 chars
	INCOME	Integer
	CREDIT_SCORE	Range(300 to 850) , Integer <= 3 Digits
	BATH_COUNT	Range(1 to 5) , Integer <= 1 Digit
	TYPE	String <= ‘EXISTING’ or ‘POTENTIAL’
	BED_COUNT	Range(1 to 5) , Integer <= 1 Digit
	SQ_FT	Integer <= 4 Digits
	RANK	Integer
	LOANAPPLICANT_ID	Integer
EXISTING	EMAIL	‘<variable chars>@<variable chars>.com’, String <=30 chars
CUSTOMERPHONE	EMAIL	‘<variable chars>@<variable chars>.com’,String <= 30 chars
	PHONE	‘(xxx)xxx-xxxx’, String <= 13 chars
POTENTIAL	EMAIL	‘<variable chars>@<variable chars>.com’, String <= 30 chars
LOANAPPLICANT	ID	5 digit Integer with Domain(10000 to 99999)
SECRETARY	EMP_ID	Integer of 12 digits Domain of EMPLOYEE.ID
	HOURLY_WAGE	Float
EMPLOYEEPHONE	EMP_ID	Integer of 12 digits Domain of EMPLOYEE.ID
	PHONE	(XXX)XXX-XXXX, String =13 Chars
LOAN	ID	10 digit Integer with Domain values(1000000000 to 9999999999)
	TERM	String <=9 Chars
	TYPE	String <=6 Chars
	DATE	MM/DD/YYYY, Date type
	CREDIT_REPORT	String <=10 Chars
	STATUS	String <=10 Chars
	LOAN_APPLICANT_ID	5 digit Integer with Domain(10000 to 99999)
LOT	SUBDIVISION_NAME	String <=25 Chars
	NUMBER	4 digit Integer with Domain(1000 to 9999)
	SIZE	Integer
	AVAILABILITY	String =1 Char, Domain('Y','N')
	FACING	String <=5 Chars, Domain('North','South','East','West')
FLOORPLAN	SUBDIVISION_NAME	String <=25 Chars
	NUMBER	3 digit Integer with Domain(100 to 999)
	FLOOR_COUNT	Integer, Domain(1,2,3)
	BED_COUNT	Integer, Domain(1,2,3,4,5)
	BATH_COUNT	Integer, Domain(1,2,3,4,5)
	NAME	String <=10 Chars

	SQ_FT	4 digit Integer with Domain(1000 to 9999)
	GARAGE_COUNT	Integer, Domain(1,2,3)
	PRICE	Integer
INVENTORY	SUBDIVISION_NAME	String <=25 Chars
	ID	5 digit Integer with Domain(10000 to 99999)
	PRICE	Integer
	MOVE_IN_DATE	MM/DD/YYYY, Date type
VISITS	CUSTOMER_EMAIL	String <=20 Chars
	SUBDIVISION_NAME	String <=25 Chars
	VISIT_DATE	MM/DD/YYYY, Date type
SIGNS	EXISTING_EMAIL	'<variable chars>@<variable chars>.com', String <= 30 chars
	CONTRACT_ID	Integer
	SUBDIVISION_NAME	String <=25 Chars
ASSOCIATES	L_SUBDIVISION_NAME	String <=25 Chars
	LOT_NUMBER	4 digit Integer with Domain(1000 to 9999)
	F_SUBDIVISION_NAME	String <=25 Chars
	FLOORPLAN_NUMBER	3 digit Integer with Domain(100 to 999)
	CONTRACT_ID	Integer
TOURS	EMP_ID	Integer of 12 digits Domain of EMPLOYEE.ID
	CUSTOMER_EMAIL	'<variable chars>@<variable chars>.com', String <= 30 chars
	I_SUBDIVISION_NAME	String <=25 Chars
	INVENTORY_ID	5 digit Integer with Domain(10000 to 99999)
	TOUR_DATE	MM/DD/YYYY, Date type
COMES	I_SUBDIVISION_NAME	String <=25 Chars
	INVENTORY_ID	5 digit Integer with Domain(10000 to 99999)
	L_SUBDIVISION_NAME	String <=25 Chars
	LOT_NUMBER	4 digit Integer with Domain(1000 to 9999)
	F_SUBDIVISION_NAME	String <=25 Chars
	FLOORPLAN_NUMBER	3 digit Integer with Domain(100 to 999)
SALESAGENT	EMPLOYEE_ID	Integer of 12 digits Domain of EMPLOYEE.ID
	SHIFT	

4. Conclusion

In this report we discussed and drew the relational schemas for Database of Home Builder Company. We also give the data type and format for each attribute in each schema. Then we explain our assumptions in the documentation. This report analyzed the logical model of Database. The next step is to implement this database. In the future, we may change some design when facing practical difficulties and other requests.