ANSWERS TO REVIEW QUESTIONS 3

1. What is the difference between a database and a table?

A table, a logical structure that represents an entity set, is only one of the components of a database. The database is a structure that contains one or more tables and metadata. The metadata are data about data. Metadata include the data (attribute) characteristics and the relationships between the entity sets.

2. What does a database expert mean when (s)he observes that a database displays both entity integrity and referential integrity?

Entity integrity describes a condition in which all tuples within a table are ***uniquely*** identified by their primary key. The unique value requirement prohibits a null primary key value, because nulls are not unique.

no duplicate or null values

Referential integrity describes a condition in which a foreign key value has a match in the corresponding table or in which the foreign key value is null.

establish relationships between tables by referencing the primary key of one table in another table.

3. Why are entity integrity and referential integrity important in a database?

Data Accuracy and Consistency:

Data Integrity and Reliability

Data Consistency across Tables

Entity integrity is important, because it means that a proper search for an ***existing*** tuple (row) will always be successful. And the failure to find a match on a row search will always mean that the row for which the search is conducted does not exist in that table. Referential integrity is important, because its existence ensures that it will be impossible to assign a non-existing foreign key value to a table. For example, the referential integrity enforcement in a SALESREP *is assigned to* CUSTOMER relationship means that it will be possible for a customer not have a sales rep assigned to him or her, but it will be impossible to assign non-existing sales rep to a customer.

4. A database manual notes that "the file contains two hundred records, each one of which contains nine fields." Use appropriate relational database terminology (table, entity set, row, tuple, entity) to "translate" the preceding statement.

Using the proper relational terminology, the statement may be translated to "the table -- or entity set -- contains two hundred rows -- or, if you like, two hundred tuples, or entities. Each of these rows contains nine attributes."

Table name: STUDENT

STU_CODE	PROF_CODE
100278	
128569	2
512272	4
531235	2
531268	
553427	1

Table name: PROFESSOR

PROF_CODE	DEPT_CODE
1	2
2	6
3	6
4	4

5. Draw the basic Entity Relationship diagram for the database shown in figure above.

6. Draw the relational schema for the database shown in figure above.
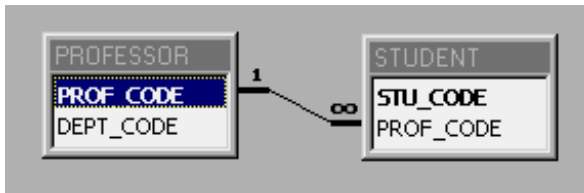
7. Suppose that you are using the following database composed of the two tables shown in figure below.

Table name: DIRECTOR

DIR_NUM	DIR_LNAME	DIR_DOB
100	Broadway	12-Jan-75
101	Hollywoody	18-Nov-63
102	Goofy	21-Jun-72

Table name: PLAY

PLAY_CODE	PLAY_NAME	DIR_NUM
1001	Cat On a Cold, Bare Roof	102
1002	Hold the Mayo, Pass the Bread	101
1003	I Never Promised You Coffee	102
1004	Silly Putty Goes To Washington	100
1005	See No Sound, Hear No Sight	101
1006	Starstruck in Biloxi	102
1007	Stranger In Parrot Ice	101

- **Identify the primary keys**

DIR_NUM is the DIRECTOR table's primary key.
PLAY_CODE is the PLAY table's primary key.

- **Identify the foreign keys**

The foreign key is DIR_NUM, located in the PLAY table. Note that the foreign key is located on the "many" side of the relationship between director and play. (Each director can direct many plays ... but each play is directed by only one director.)

- **Draw the entity relationship model**

- **Draw the relational schema to show the relationship between DIRECTOR and PLAY.**

ANSWERS TO PROBLEMS 3
PART 1. Use the database shown in figure below to answer problems 1 through 9.

Table name: EMPLOYEE

EMP_CODE	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	STORE_CODE
1	Mr	Govender	Adimoolam	W	21-May-70	3
2	Ms	Ratula	Nancy		09-Feb-75	2
3	Ms	Greenboro	Lottie	R	02-Oct-67	4
4	Mrs	Rumpersfro	Jennie	S	01-Jun-77	5
5	Mr	Smith	Robert	L	23-Nov-65	3
6	Mr	Renselaer	Cary	A	25-Dec-71	1
7	Mr	Ogallo	Roberto	S	31-Jul-68	3
8	Ms	Van Blerk	Elandri	I	10-Sep-74	1
9	Mr	Eindsmar	Jack	W	19-Apr-61	2
10	Mrs	Jones	Rose	R	06-Mar-72	4
11	Mr	Broderick	Tom		21-Oct-78	3
12	Mr	Washington	Alan	Y	08-Sep-80	2
13	Mr	Smith	Peter	N	25-Aug-70	3
14	Ms	Smith	Sherry	H	25-May-72	4
15	Mr	Olenko	Howard	U	24-May-70	5
16	Mr	Archialo	Barry	V	03-Sep-66	5
17	Ms	Grimaldo	Jeanine	K	12-Nov-76	4
18	Mr	Rosenberg	Andrew	D	24-Jan-77	4
19	Mr	Bophela	Ingwe	F	03-Oct-74	4
20	Mr	Mckee	Robert	S	06-Mar-76	1
21	Ms	Baumann	Jennifer	A	11-Dec-80	3

Table name: STORE

STORE_CODE	STORE_NAME	STORE_YTD_SALES	REGION_CODE	EMP_CODE
1	Access Junction	€792 730.05	2	8
2	Database Corner	€1 123 370.04	2	12
3	Tuple Charge	€779 558.74	1	7
4	Attribute Alley	€746 209.16	2	3
5	Primary Key Point	€2 314 777.78	1	15

Table name: REGION

REGION_CODE	REGION_DESCRIPT
1	East
2	West

1. **For each table, identify the primary key and the foreign key(s). If a table does not have a foreign key, write NONE in the assigned space.**

Table	Primary key	Foreign Key(s)
EMPLOYEE	EMP_CODE	STORE_CODE
STORE	STORE_CODE	REGION_CODE, EMP_CODE
REGION	REGION_CODE	NONE

2. **Do the tables exhibit entity integrity? Answer Yes or No, then explain your answer.**

Table	Entity Integrity?	Explanation
EMPLOYEE	Yes	Each EMP_CODE value is unique and there are no nulls
STORE	Yes	Each STORE_CODE value is unique and there are no nulls
REGION	Yes	Each REGION_CODE value is unique and there are no nulls

3. **Do the tables exhibit referential integrity? Answer Yes or No, then explain your answer. Write NA (Not Applicable) if the table does not have a foreign key.**

| Table	Referential Integrity?	Explanation
EMPLOYEE	Yes	Each STORE_CODE value in EMPLOYEE points to an *existing* STORE_CODE value in STORE.
STORE	Yes	Each REGION_CODE value in STORE points to an *existing* REGION_CODE value in REGION and each EMP_CODE value in STORE points to an *existing* EMP_CODE value in EMPLOYEE.
REGION	NA	

4. **Describe the type(s) of relationship(s) between STORE and REGION.**

Because REGION_CODE values occur more than once in STORE, we may conclude that each REGION can contain many stores. But, since each STORE is located in only one REGION, the relationship between STORE and REGION is M:1. (It is, of course, equally true that the relationship between REGION and STORE is 1:M.)

5. **Draw the Entity Relationship diagram for the relationship between STORE and REGION.**

6. **Draw the Relational Schema for the relationship between STORE and REGION.**
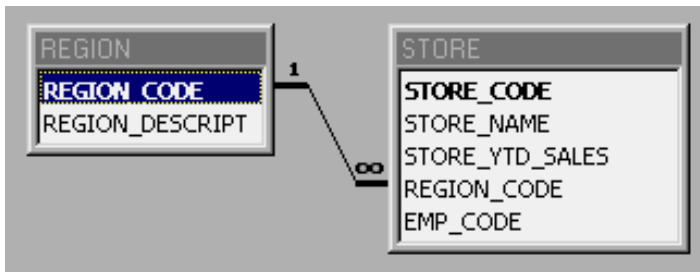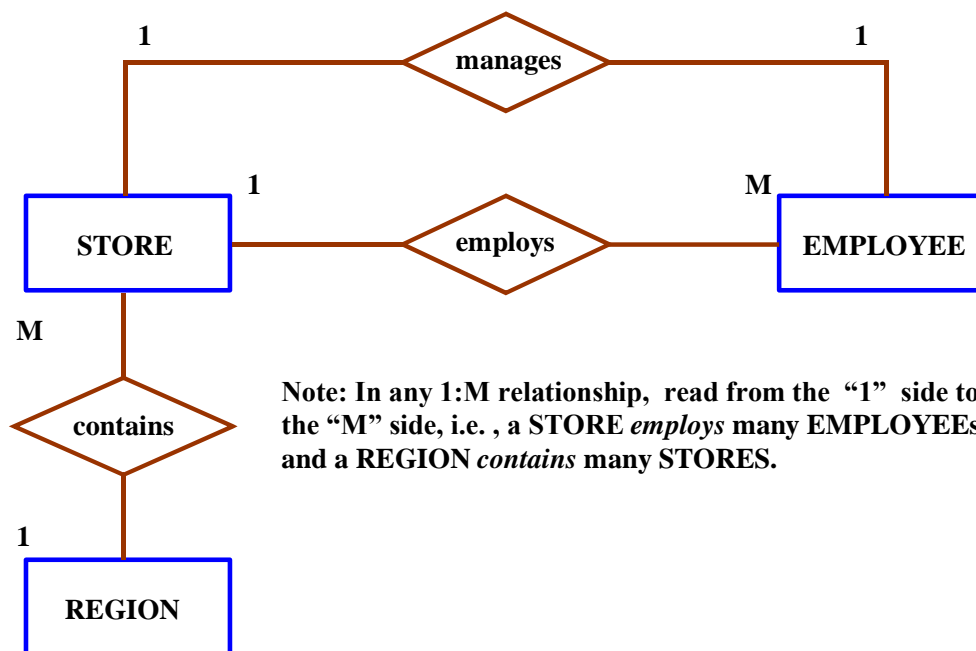

7. **Describe the type(s) of relationship(s) between EMPLOYEE and STORE. (*Hint*: Each store employs many employees, one of whom manages the store.)**

There are TWO relationships between STORE and EMPLOYEE. The first relationship, expressed by STORE *employs* EMPLOYEE, is a 1:M relationship, because one store can employ many employees and each employee is

employed by one store. The second relationship, expressed by EMPLOYEE *manages* STORE, is a 1:1 relationship, because each store is managed by one employee and an employee manages only one store.

> **Note: It is useful to introduce several ways in which the *manages* relationship may be implemented. For example, rather than creating the *manages* relationship between EMPLOYEE and STORE, it is possible to simply list the manager's name as an attribute in the STORE table. This approach creates a redundancy which may not do much damage if the information requirements are limited. However, if it is necessary to keep track of each manager's sales and personnel management performance by store, the manages relationship we have shown here will do a much better job in terms of information generation.**

8. **Draw the Entity Relationship diagram to show the relationships among EMPLOYEE, STORE, and REGION.**

> **Note: In any 1:M relationship, read from the "1" side to the "M" side, i.e. , a STORE *employs* many EMPLOYEEs and a REGION *contains* many STORES.**

9. **Draw the Relational Schema to show the relationships between EMPLOYEE, STORE, and REGION.**
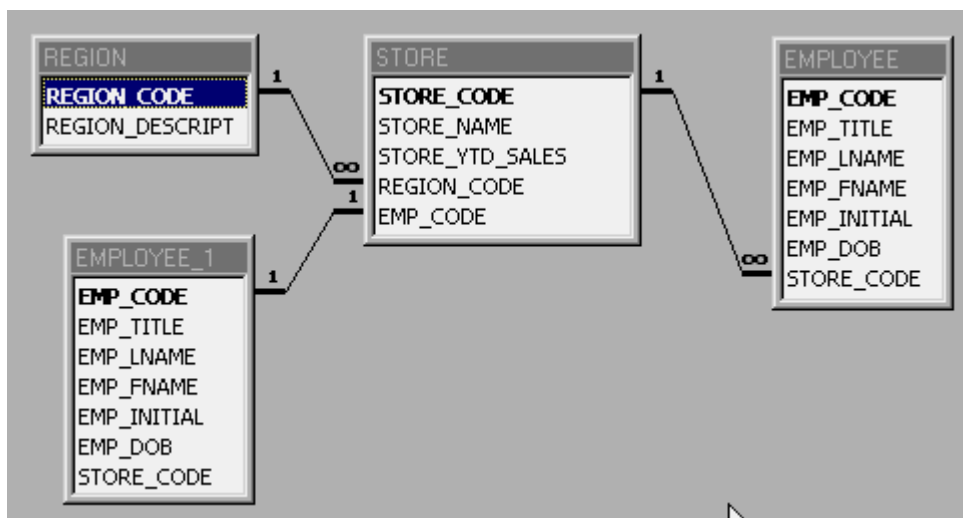
Note: The relational schema was generated in Microsoft Access. If a relationship already exists between two entities, Access generates a virtual table (in this case, EMPLOYEE_1) to generate the additional relationship. The virtual table cannot be queried; its only function is to store the *manages* relationship between EMPLOYEE and STORE. Just how multiple relationships are stored and managed is a function of the software you use.

FIGURE P2.1 The DealCo class ERD
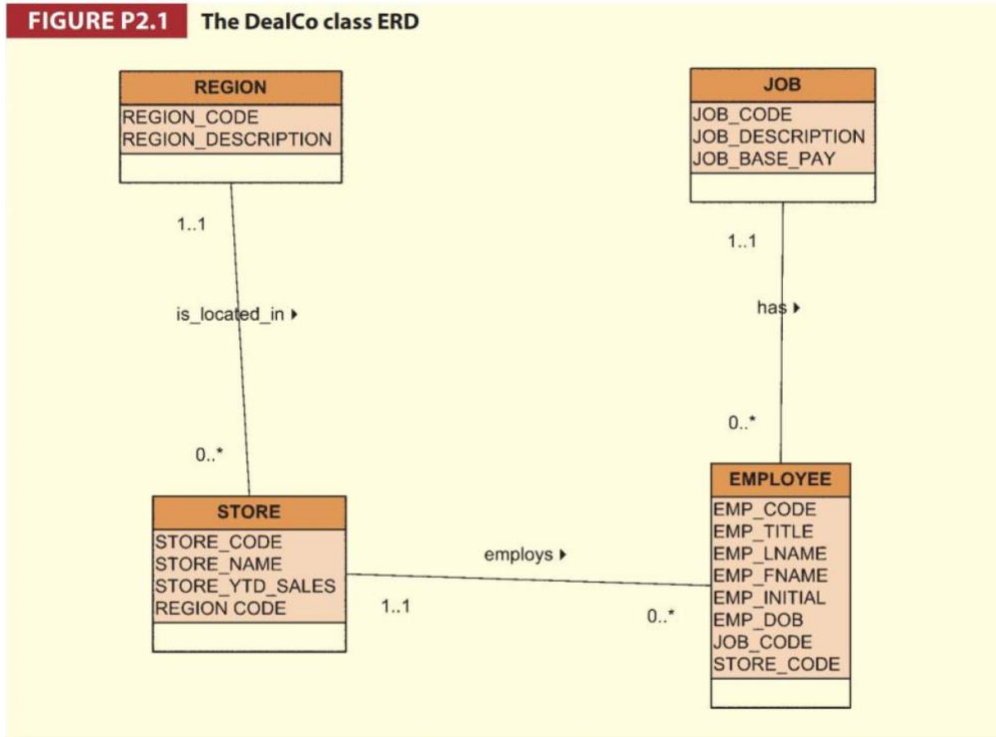
PART 2. Use the database shown in figure below to answer problems 1 through 5.

Table name: PRODUCT   Primary key: PROD_CODE

Foreign key: VEND_CODE

PROD_CODE	PROD_DESCRIPTION	PROD_STOCK_DATE	PROD_ON_HAND	PROD_PRICE	VEND_CODE
12-WW/P2	18 cm power saw blade	07-Apr-16	12	10.94	123
1QQ23-55	6 cm wood screw, 100	19-Mar-16	123	13.55	123
231-78-W	PVC pipe, 8 cm, 2.44 m	07-Dec-15	45	17.01	121
33564/U	Rat-tail file, 0.5 cm, fine	08-Mar-16	18	10.94	123
AR/3/TYR	Cordless drill, 0.6 cm	29-Nov-15	8	136.33	121
DT-34-WW	Philips screwdriver pack	20-Dec-15	11	118.40	123
EE3-67/W	Sledge hammer, 7 kg	25-Feb-16	9	114.21	121
ER-56/DF	Houselite chain saw, 40 cm	28-Dec-15	7	1186.04	125
FRE-TRY9	Jigsaw, 30 cm blade	12-Aug-15	67	11.15	125
SE-67-89	Jigsaw, 20 cm blade	11-Oct-15	34	11.07	125
ZW-QR/AV	Hardware cloth, 0.6 cm.	23-Apr-16	14	110.26	123
ZX-WR/FR	Claw hammer	01-Mar-16	15	17.07	121

Table name: VENDOR   Primary key: VEND_CODE

Foreign key: none

VEND_CODE	VEND_NAME	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
120	Bargain Snapper, Inc.	Melanie T. Travis	0181	899-1234
121	Cut 'n' Glow Co.	Henry J. Olero	0181	342-9896
122	Rip & Rattle Supply Co.	Anne R. Morrins	0113	225-1127
123	Tools 'R' Us	Juliette G. McHenry	0161	546-7894
124	Trowel & Dowel, Inc.	George F. Frederick	0113	453-4567
125	Bow & Wow Tools	Bill S. Sedwick	0113	324-9988

1. **For each table, identify the primary key and the foreign key(s). If a table does not have a foreign key, write NONE in the assigned space.**

Table	Primary key	Foreign Key(s)
PRODUCT	PROD_CODE	VEND_CODE
VENDOR	VEND_CODE	NONE

2. **Do the tables exhibit entity integrity? Answer Yes or No, then explain your answer.**

Table	Entity Integrity?	Explanation
PRODUCT	Yes	All the PROD_CODE values in PRODUCT are unique and there are no nulls.
VENDOR	Yes	All the VEND_CODE values in VENDOR are unique and there are no nulls.

3. **Do the tables exhibit referential integrity? Answer Yes or No, then explain your answer. Write NA (Not Applicable) if the table does not have a foreign key.**

Table	Referential Integrity?	Explanation
PRODUCT	Yes	All the VEND_CODE values in PRODUCT point to existing VEND_CODE values in VENDOR
VENDOR	NA	

4. **Draw the Entity Relationship diagram for this database.**

Note: In this case, we assume that each product is supplied by a single vendor. It is, of course, possible for a situation to exist in which a vendor can supply many products and each product can be supplied by many vendors. In this latter scenario, there exists a M:N relationship between VENDOR and PRODUCT. This M:N relationship must then be decomposed into two sets of 1:M relationships, thus requiring the creation of a third table structure (a composite entity) to form the "bridge" between VENDOR and PRODUCT. We will learn how to decompose M:N relationships into two 1:M relationships in the following lectures.

5. **Draw the Relational Schema for this database.**
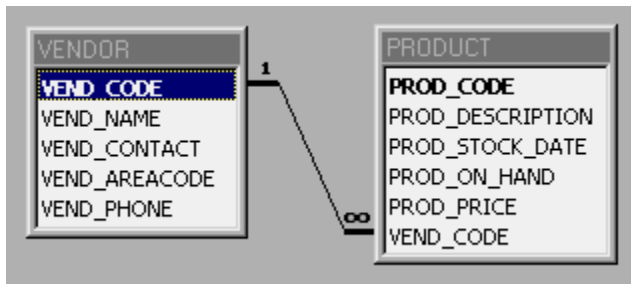
PART 3. Use the database shown in figure below to answer problems 1 through 5.

Table name: TRUCK — Primary key: TRUCK_NUM

Foreign key: BASE-CODE, TYPE_CODE

TRUCK_ NUM	BASE_ CODE	TYPE_ CODE	TRUCK_ KM	TRUCK_BUY_ DATE	TRUCK_SERIAL_ NUM
1001	501	1	32 123.50	23-Sep-13	AA-322-12212-W11
1002	502	1	76 984.30	05-Feb-12	AC-342-22134-Q23
1003	501	2	12 346.60	11-Nov-13	AC-445-78656-Z99
1004		1	2 894.30	06-Jan-14	WQ-112-23144-T34
1005	503	2	45 673.10	01-Mar-13	FR-998-32245-W12
1006	501	2	193 245.70	15-Jul-10	AD-456-00845-R45
1007	502	3	32 012.30	17-Oct-11	AA-341-96573-Z84
1008	502	3	44 213.60	07-Aug-12	DR-559-22189-D33
1009	503	2	10 932.90	12-Feb-14	DE-887-98456-E94

Table name: BASE — Primary key: BASE_CODE

Foreign key: none

BASE_CODE	BASE_CITY	BASE_PROVINCE	BASE_AREA_CODE	BASE_ PHONE	BASE_MANAGER
501	Polokwane	Limpopo	0700	123-4567	Sibusiso Balisa
502	Cape Town	Western Cape	7100	234-5678	Clementine Daniels
503	Best	North Brabant	4567	345-6789	Maria J. Talindo
504	Durban	KwaZulu-Natal	4001	456-7890	Pragasen Khan

Table name: TYPE — Primary key: TYPE_CODE

Foreign key: none

TYPE_CODE	TYPE_DESCRIPTION
1	Single box, double-axle
2	Single box, single-axle
3	Tandem trailer, single-axle

1. **For each table, identify the primary key and the foreign key(s). If a table does not have a foreign key, write NONE in the assigned space.**

Table	Primary key	Foreign Key(s)
TRUCK	TRUCK_NUM *	BASE_CODE, TYPE_CODE
BASE	BASE_CODE	NONE
TYPE	TYPE_CODE	NONE

> Note: The TRUCK_SERIAL_NUM could also be designated as the primary key. But we prefer to design TRUCK_NUM as the primary key.
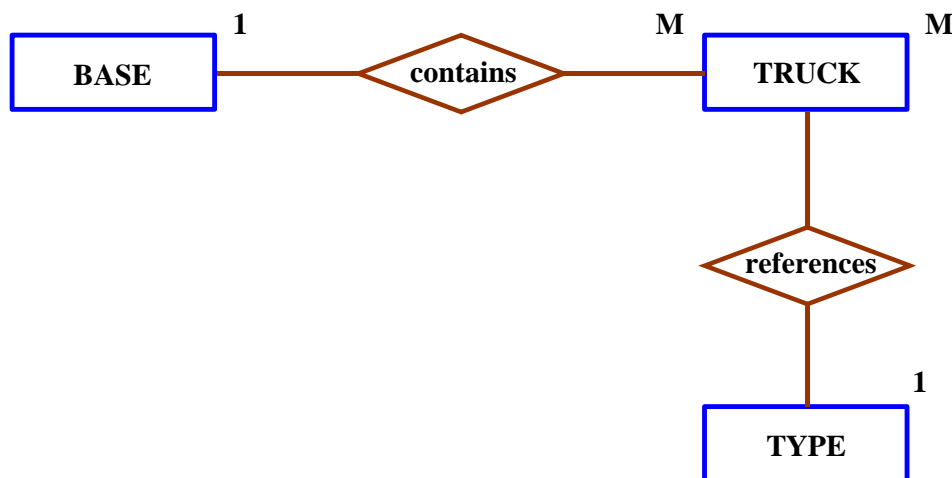
2. **Do the tables exhibit entity integrity? Answer Yes or No, then explain your answer.**

Table	Entity Integrity?	Explanation
TRUCK	Yes	The TRUCK_NUM values in the TRUCK table are all unique and there are no nulls.
BASE	Yes	The BASE_CODE values in the BASE table are all unique and there are no nulls.
TYPE	Yes	The TYPE_CODE values in the TYPE table are all unique and there are no nulls.

8

3. Do the tables exhibit referential integrity? Answer Yes or No, then explain your answer. Write NA (Not Applicable) if the table does not have a foreign key.

Table	Referential Integrity?	Explanation
TRUCK	Yes	The BASE_CODE values in the TRUCK table reference existing BASE_CODE values in the BASE table or they are null. (The TRUCK table's BASE_CODE is null for TRUCK_NUM = 1004.) Also, the TYPE_CODE values in the TRUCK table reference existing TYPE_CODE values in the TYPE table.
BASE	NA	
TYPE	NA	

4. Draw the Entity Relationship diagram for this database.

BASE — 1 — contains — M — TRUCK — M — references — 1 — TYPE

5. Draw the Relational Schema for this database.

BASE
- BASE_CODE
- BASE_CITY
- BASE_STATE
- BASE_AREA_CODE
- BASE_PHONE
- BASE_MANAGER

TRUCK
- TRUCK_NUM
- BASE_CODE
- TYPE_CODE
- TRUCK_MILES
- TRUCK_BUY_DATE
- TRUCK_SERIAL_NUM

TYPE
- TYPE_CODE
- TYPE_DESCRIPTION