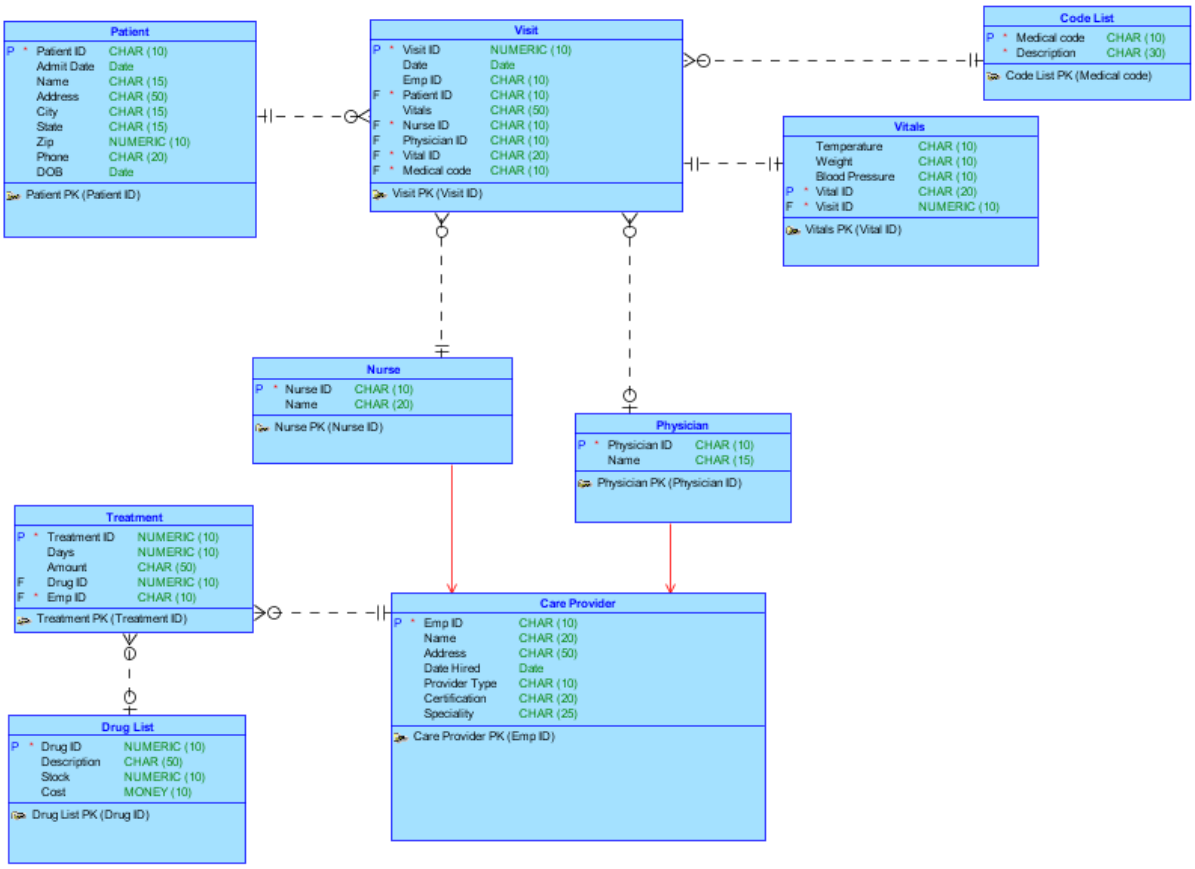
Date: 10/14/2016

Inventory Management System

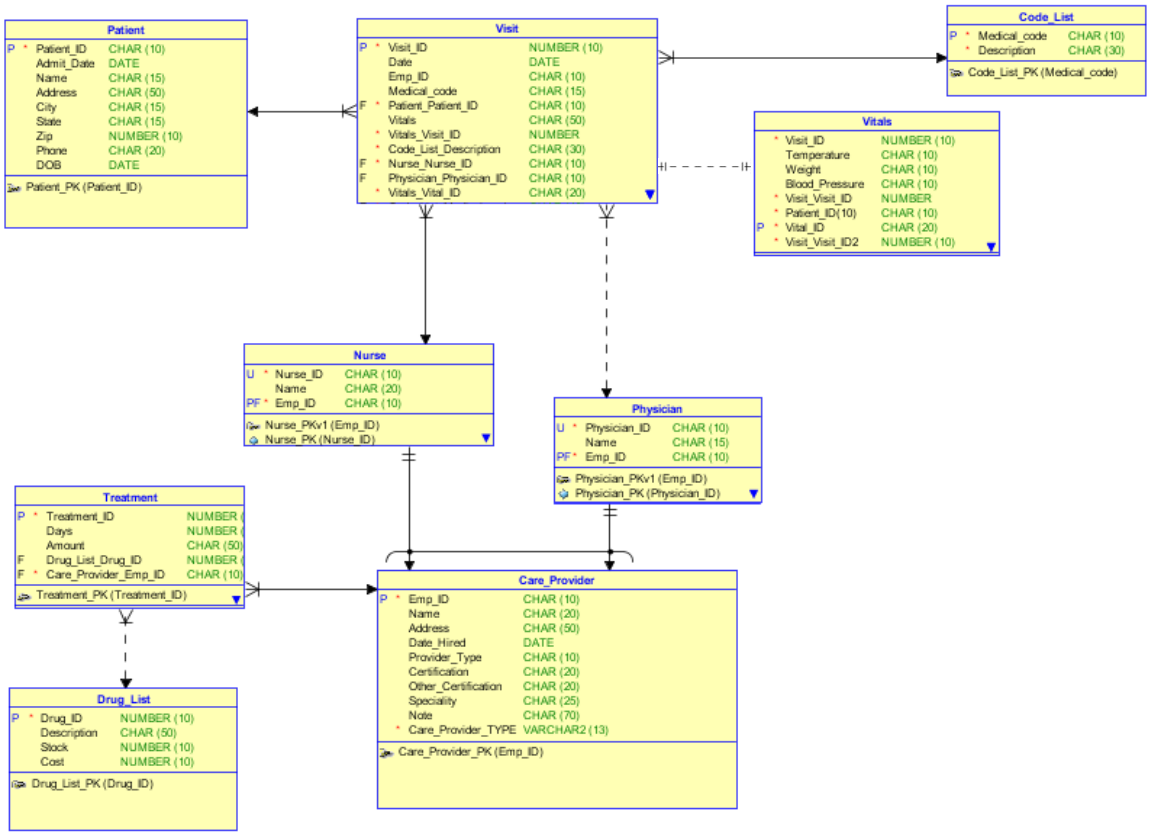
- Designing a database for physical inventory management in business
- Creating entities and related tables to store inventory data
- Engineering ER diagram based on assumptions
- Generating relational model using Oracle Database Developer
- Generating DDL to implement database
- Populating and tested data using SQL queries for reporting

ER Diagram



Date: 10/14/2016

Relational Model



Date: 10/14/2016

DDL

```
-- Generated by Oracle SQL Developer Data Modeler 4.1.3.901
--   at:       2016-10-16 23:21:16 EDT
--   site:     Oracle Database 11g
--   type:     Oracle Database 11g




CREATE TABLE Care_Provider
 (
   Emp_ID           CHAR (10) NOT NULL ,
   Name             CHAR (20) ,
   Address          CHAR (50) ,
   Date_Hired       DATE ,
   Provider_Type    CHAR (10) ,
   Certification    CHAR (20) ,
   Speciality       CHAR (25) ,
   Care_Provider_TYPE VARCHAR2 (13) NOT NULL
 ) ;
ALTER TABLE Care_Provider ADD CONSTRAINT CH_INH_Care_Provider CHECK ( Care_Provider_TYPE IN
('Care_Provider', 'Nurse', 'Physician')) ;

ALTER TABLE Care_Provider ADD CONSTRAINT Care_Provider_PK PRIMARY KEY ( Emp_ID ) ;



CREATE TABLE Code_List
 (
   Medical_code CHAR (10) NOT NULL ,
   Description  CHAR (30) NOT NULL
```

```
 ) ;
ALTER TABLE Code_List ADD CONSTRAINT Code_List_PK PRIMARY KEY ( Medical_code ) ;


CREATE TABLE Drug_List
 (
   Drug_ID     NUMBER (10) NOT NULL ,
   Description CHAR (50) ,
   Stock       NUMBER (10) ,
   Cost        NUMBER (10)
 ) ;
ALTER TABLE Drug_List ADD CONSTRAINT Drug_List_PK PRIMARY KEY ( Drug_ID ) ;


CREATE TABLE Nurse
 (
   Emp_ID   CHAR (10) NOT NULL ,
   Nurse_ID CHAR (10) NOT NULL ,
   Name     CHAR (20)
 ) ;
ALTER TABLE Nurse ADD CONSTRAINT Nurse_PK PRIMARY KEY ( Emp_ID ) ;
ALTER TABLE Nurse ADD CONSTRAINT Nurse_PKv1 UNIQUE ( Nurse_ID ) ;


CREATE TABLE Patient
 (
   Patient_ID CHAR (10) NOT NULL ,
   Admit_Date DATE ,
   Name       CHAR (15) ,
   Address    CHAR (50) ,
```

```
  City      CHAR (15) ,
  State     CHAR (15) ,
  Zip       NUMBER (10) ,
  Phone     CHAR (20) ,
  DOB       DATE
 ) ;
ALTER TABLE Patient ADD CONSTRAINT Patient_PK PRIMARY KEY ( Patient_ID ) ;


CREATE TABLE Physician
 (
  Emp_ID      CHAR (10) NOT NULL ,
  Physician_ID CHAR (10) NOT NULL ,
  Name        CHAR (15)
 ) ;
ALTER TABLE Physician ADD CONSTRAINT Physician_PK PRIMARY KEY ( Emp_ID ) ;
ALTER TABLE Physician ADD CONSTRAINT Physician_PKv1 UNIQUE ( Physician_ID ) ;


CREATE TABLE Treatment
 (
  Treatment_ID        NUMBER (10) NOT NULL ,
  Days                NUMBER (10) ,
  Amount              CHAR (50) ,
  Drug_List_Drug_ID   NUMBER (10) ,
  Care_Provider_Emp_ID CHAR (10) NOT NULL
 ) ;
ALTER TABLE Treatment ADD CONSTRAINT Treatment_PK PRIMARY KEY ( Treatment_ID ) ;
```

Date: 10/14/2016

```
CREATE TABLE Visit
 (
   Visit_ID             NUMBER (10) NOT NULL ,
   "Date"               DATE ,
   Emp_ID               CHAR (10) ,
   Patient_Patient_ID   CHAR (10) NOT NULL ,
   Vitals               CHAR (50) ,
   Nurse_Nurse_ID       CHAR (10) NOT NULL ,
   Physician_Physician_ID CHAR (10) ,
   Vitals_Vital_ID      CHAR (20) NOT NULL ,
   Code_List_Medical_code CHAR (10) NOT NULL
 ) ;
CREATE UNIQUE INDEX Visit__IDX ON Visit
 (
   Vitals_Vital_ID ASC
 )
 ;
ALTER TABLE Visit ADD CONSTRAINT Visit_PK PRIMARY KEY ( Visit_ID ) ;


CREATE TABLE Vitals
 (
   Temperature    CHAR (10) ,
   Weight         CHAR (10) ,
   Blood_Pressure CHAR (10) ,
   Vital_ID       CHAR (20) NOT NULL ,
   Visit_Visit_ID NUMBER (10) NOT NULL
 ) ;
CREATE UNIQUE INDEX Vitals__IDX ON Vitals
 (
```

```
   Visit_Visit_ID ASC
 )
 ;
ALTER TABLE Vitals ADD CONSTRAINT Vitals_PK PRIMARY KEY ( Vital_ID ) ;
```

```
ALTER TABLE Nurse ADD CONSTRAINT Nurse_Care_Provider_FK FOREIGN KEY ( Emp_ID ) REFERENCES
Care_Provider ( Emp_ID ) ;
```

```
ALTER TABLE Physician ADD CONSTRAINT Physician_Care_Provider_FK FOREIGN KEY ( Emp_ID )
REFERENCES Care_Provider ( Emp_ID ) ;
```

```
ALTER TABLE Treatment ADD CONSTRAINT Treatment_Care_Provider_FK FOREIGN KEY (
Care_Provider_Emp_ID ) REFERENCES Care_Provider ( Emp_ID ) ;
```

```
ALTER TABLE Treatment ADD CONSTRAINT Treatment_Drug_List_FK FOREIGN KEY ( Drug_List_Drug_ID )
REFERENCES Drug_List ( Drug_ID ) ;
```

```
ALTER TABLE Visit ADD CONSTRAINT Visit_Code_List_FK FOREIGN KEY ( Code_List_Medical_code )
REFERENCES Code_List ( Medical_code ) ;
```

```
ALTER TABLE Visit ADD CONSTRAINT Visit_Nurse_FK FOREIGN KEY ( Nurse_Nurse_ID ) REFERENCES
Nurse ( Nurse_ID ) ;
```

```
ALTER TABLE Visit ADD CONSTRAINT Visit_Patient_FK FOREIGN KEY ( Patient_Patient_ID ) REFERENCES
Patient ( Patient_ID ) ;
```

```
ALTER TABLE Visit ADD CONSTRAINT Visit_Physician_FK FOREIGN KEY ( Physician_Physician_ID )
REFERENCES Physician ( Physician_ID ) ;
```

```
ALTER TABLE Visit ADD CONSTRAINT Visit_Vitals_FK FOREIGN KEY ( Vitals_Vital_ID ) REFERENCES Vitals (
Vital_ID ) ;
```

```
ALTER TABLE Vitals ADD CONSTRAINT Vitals_Visit_FK FOREIGN KEY ( Visit_Visit_ID ) REFERENCES Visit ( Visit_ID ) ;


CREATE OR REPLACE TRIGGER ARC_FKArc_1_Physician BEFORE
  INSERT OR
  UPDATE OF Emp_ID ON Physician FOR EACH ROW DECLARE d VARCHAR2 (13);
  BEGIN
    SELECT A.Care_Provider_TYPE
    INTO d
    FROM Care_Provider A
    WHERE A.Emp_ID = :new.Emp_ID;
    IF (d        IS NULL OR d <> 'Physician') THEN
      raise_application_error(-20223,'FK Physician_Care_Provider_FK in Table Physician violates Arc constraint on Table Care_Provider - discriminator column Care_Provider_TYPE doesn''t have value ''Physician''');
    END IF;
  EXCEPTION
  WHEN NO_DATA_FOUND THEN
    NULL;
  WHEN OTHERS THEN
    RAISE;
  END;
  /
CREATE OR REPLACE TRIGGER ARC_FKArc_1_Nurse BEFORE
  INSERT OR
  UPDATE OF Emp_ID ON Nurse FOR EACH ROW DECLARE d VARCHAR2 (13);
  BEGIN
    SELECT A.Care_Provider_TYPE
    INTO d
    FROM Care_Provider A
    WHERE A.Emp_ID = :new.Emp_ID;
```

```
   IF (d        IS NULL OR d <> 'Nurse') THEN

     raise_application_error(-20223,'FK Nurse_Care_Provider_FK in Table Nurse violates Arc constraint on
Table Care_Provider - discriminator column Care_Provider_TYPE doesn''t have value ''Nurse''');

   END IF;

 EXCEPTION

 WHEN NO_DATA_FOUND THEN

   NULL;

 WHEN OTHERS THEN

   RAISE;

 END;

 /



-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE                         9

-- CREATE INDEX                         2

-- ALTER TABLE                         22

-- CREATE VIEW                          0

-- ALTER VIEW                          0

-- CREATE PACKAGE                       0

-- CREATE PACKAGE BODY                  0

-- CREATE PROCEDURE                     0

-- CREATE FUNCTION                      0

-- CREATE TRIGGER                       2

-- ALTER TRIGGER                       0

-- CREATE COLLECTION TYPE               0

-- CREATE STRUCTURED TYPE               0

-- CREATE STRUCTURED TYPE BODY          0

-- CREATE CLUSTER                       0
```

-- CREATE CONTEXT 0

-- CREATE DATABASE 0

-- CREATE DIMENSION 0

-- CREATE DIRECTORY 0

-- CREATE DISK GROUP 0

-- CREATE ROLE 0

-- CREATE ROLLBACK SEGMENT 0

-- CREATE SEQUENCE 0

-- CREATE MATERIALIZED VIEW 0

-- CREATE SYNONYM 0

-- CREATE TABLESPACE 0

-- CREATE USER 0

--

-- DROP TABLESPACE 0

-- DROP DATABASE 0

--

-- REDACTION POLICY 0

--

-- ORDS DROP SCHEMA 0

-- ORDS ENABLE SCHEMA 0

-- ORDS ENABLE OBJECT 0

--

-- ERRORS 0

-- WARNINGS 0

Populate Sample Data

Date: 10/14/2016

Five patients (including a patient with the same name as yourself), five nurses and 4 physicians, 4 visit types, five drugs and 15 visits

Drug list

	DRUG_ID	DESCRIPTION	STOCK_O...	COST_PE...
1	631	Robitussin 80 CC ...	49	3
2	578	Claritin 56 milligram ...	720	0.1
3	443	Loratadine 90 milligram ...	290	0.12
4	375	Zyrtec 30 milligram ...	880	0.15
5	374	Aspirin 81 milligram ...	1052	0.1

Nurse

	EMP_ID	NURSE_ID	NAME
1	A34520	A345231	Jack Sparrow, RN
2	A37946	A379462	Goldman Sachs, RN
3	A02849	A028493	Bob Dylan
4	A03928	A039289	Morry Bi, RN
5	A38947	A389470	Shirly Lu, RN

Physician

	EMP_ID	PHYSICIA...	NAME
1	A22035	A220354	Marla Myles, MD
2	A37895	A378956	Aaron Wei, MD
3	A89379	A893793	Abbey Chieng
4	A38496	A384961	Brimidicle Yong

Code list

	MEDICAL...	DESCRIPTION
1	MN302.1	Pneumonia
2	XN632.09	Flu
3	RD603.33	Diarrhea
4	MN332.1	Bronchitis

Vitals (15 visits)

	VISIT_ID	TEMPERA...	WEIGHT	BLOOD_P...	VISIT_VISIT_ID
1	1002	103F	100Lb	132/85	1002
2	1003	102F	102Lb	131/86	1003
3	1001	102F	110Lb	133/90	1001
4	1004	102F	112Lb	135/88	1004
5	1005	103F	108Lb	133/89	1005
6	1006	102F	108Lb	132/88	1006
7	1007	100F	105Lb	131/87	1007
8	1008	102F	104Lb	132/86	1008
9	1009	103F	106Lb	142/89	1009
10	1010	102F	102Lb	132/78	1010
11	1011	101F	100Lb	130/76	1011
12	1012	100F	99Lb	128/78	1012
13	1013	102F	100Lb	129/78	1013
14	1014	101F	100Lb	130/90	1014
15	1015	102F	99Lb	129/89	1015

Date: 10/14/2016

Query:
Natural language statement: List all the patients
SQL statement:

```
SELECT *

FROM PATIENT;
```

Query Result

All Rows Fetched: 5 in 0.015 seconds

	PATIENT_ID	ADMIT_DATE	NAME	ADDRESS	CITY	STATE	ZIP	TELEPHONE	DOB
1	Ars1	17-MAY-16	Shimeng Lin	216 W 108th Street, Apt 9B	New York	New York	10025	(765)714-8976	12-OCT-90
2	Ars2	03-OCT-16	Barby May	216 W 109th Street, Apt 2B	New York	New York	10025	(317)714-8976	01-OCT-05
3	Ars3	04-OCT-16	Gill Brown	216 W 102th Street, Apt 2B	New York	New York	10025	(317)714-8970	16-OCT-16
4	Ars4	18-MAY-00	Darcy Bi	216 W 102th Street, Apt 3C	New York	New York	10025	(317)714-8070	11-OCT-03
5	Ars 5	11-OCT-16	Mary McQueen	206 W 102th Street, Apt 3C	New York	New York	10025	(317)718-8070	16-OCT-16

Query:
Natural language statement: List patient ID and name of each patient and number of visits the patient has made.
SQL statement:

```
SELECT
    PATIENT.PATIENT_ID,
    PATIENT.NAME,
    COUNT(VISIT.PATIENT_ID) AS Cnt
FROM
     PATIENT,
     VISIT
WHERE
     PATIENT.PATIENT_ID = VISIT.PATIENT_ID
GROUP BY
    PATIENT.PATIENT_ID,
    PATIENT.NAME;
```

Query:
Natural language statement: List the drug that has been prescribed most often.
SQL statement:

```
SELECT
    DRUG_LIST.DRUG_ID,
    TREATMENT.DRUG_LIST_DRUG_ID,
    COUNT(TREATMENT.DRUG_LIST_DRUG_ID) AS Cnt
FROM
     DRUG_LIST,
     TREATMENT
```

```
WHERE
    TREATMENT.DRUG_LIST_DRUG_ID = DRUG_LIST.DRUG_ID
GROUP BY
    DRUG_LIST.DRUG_ID,
    TREATMENT.DRUG_LIST_DRUG_ID;
```

Query:
Natural language statement: Identify the number of visits handled by each nurse.
SQL statement:

```
SELECT
    VISIT.VISIT_ID,
    NURSE.NURSE_ID
    COUNT(VISIT.VISIT_ID) AS Cnt
FROM
    PATIENT,
    VISIT
WHERE
    PATIENT.PATIENT_ID = VISIT.PATIENT_ID
GROUP BY
    VISIT.VISIT_ID,
    NURSE.NURSE_ID;
```

Query:
Natural language statement: Identify all the physicians that a patient has seen.
SQL statement:

```
SELECT
    PHYSICIAN.PHYSICIAN_ID,
    PHYSICIAN.NAME,
    PATIENT.PATIENT_ID
FROM
    PATIENT,
    PHYSICIAN
WHERE
    PATIENT.PHYSICIAN_ID = PHYSICIAN.PHYSICIAN_ID;
```