2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023

The Harbor of Engineering
Education for 130 Years

ASEE

Paper ID #38858

Rule-Based Database System for Airplane Maintenance Project

Dr. Reza Sanati-Mehrizy, Utah Valley University

Reza Sanati-Mehrizy is a professor of Computer Science Department at Utah Valley University, Orem, Utah. He received his M.S. and Ph.D. in Computer Science from the University of Oklahoma, Norman, Oklahoma. His research focuses on diverse areas such as: D

Cody Lance Strange
Dr. Afsaneh Minaie, Utah Valley University

Afsaneh Minaie is a Professor of Electrical and Computer Engineering at Utah Valley University. She received her B.S., M.S., and Ph.D. all in Electrical Engineering from the University of Oklahoma. Her research interests include gender issues in the academic sci

Rule Based Database System for Airplane Maintenance

Reza Sanati-Mehrizy — **Professor**
Cody Strange — **Student**
Afsaneh Minaie — **Professor**

College of Engineering and Technology
Utah Valley University

Abstract

Organizations have many business rules (constraints) to implement in their daily operations. This is done mainly by action assertions traditionally implemented in procedural logic buried deeply within user's application program in a form that is virtually unrecognizable, unmanageable, and inconsistent[1]. This approach places a heavy burden on the programmer, who must know all the constraints that an action may violate and must include checks for each of these constraints. An omission, misunderstanding, or error by the programmer will likely leave the database in an inconsistent state.

A more modern approach is to define assertions at a conceptual level without specifying how the rule will be implemented. Thus, there needs to be a specification language for business rules. Entity Relationship (ER) model is a common conceptual database design tool used for relational database design. To enforce the business rules, some business rules can be included in this ER/EER model in the form of constraints. This inclusion becomes a good reminder for the programmer to include them in his database implementation.

Some constraints can be enforced as column constraint or table constraint while tables are being created. Complicated business rules can't be implemented this way. Writing trigger is a good way of implementing these complicated business rules. Trigger is a program and is more flexible than a constraint. Trigger has Event, Condition and Action (ECA) property. When an event takes place and consequently a condition becomes true, the trigger acts automatically and modifies the database as needed. To write trigger, there are options of before and after. The option before means before the operation is performed, the trigger will be executed to make sure it is ok for the operation to be performed. If this is the case, the operation will be executed otherwise the operation will not be executed.

In this paper, we use the ER notation to represent some business rules (constraints) graphically for airplane maintenance database system and write triggers to implement them to ensure services provided for the airplanes only by well-trained mechanics. This is a continuation of the work done by the same authors[4]

Introduction

Business rules define or constrain different aspects of business. By applying a business rule, it is intended to assert business structure, or to control or influence the behavior and daily operation of the business[2]. Organizations have many business rules to implement in their daily operations. Traditionally, this is done mainly by action assertions implemented in user's application programs in a form that is not clearly recognizable, manageable, and consistent. This approach places a heavy burden on the programmer to know all the constraints that an action may violate, implement them carefully and include check for each of these constraints. This is not a reliable approach because an omission, misunderstanding, or error by the programmer will likely leave the database in an inconsistent state.

The more modern and more reliable approach is to define these assertions at a conceptual level by including them in the ER/EER model without specifying how the rules will be implemented. This approach builds the constraints in the system to reduce the possibility of making errors by the programmer. Thus, there needs to be a specification language for expressing business rules. We have seen the Entity Relationship (ER) and Enhanced Entity Relationship (EER) notations work well for specifying many business rules. In fact, EER notation was invented to allow more business rules to be shown in graphical form than the simpler ER notation[1].

In this paper, we use the ER notation to represent business rules graphically at conceptual level for a relational data model to enforce database consistency for the type of services a mechanic can provide for an airplane.

Project Description

In this project, we have simplified the airplane maintenance system by including the entity types Airplane, Mechanic, Service and Training and the relationships Requires, Receives and Provides. Figure 1 is an Entity Relationship diagram that represents this simple database model.

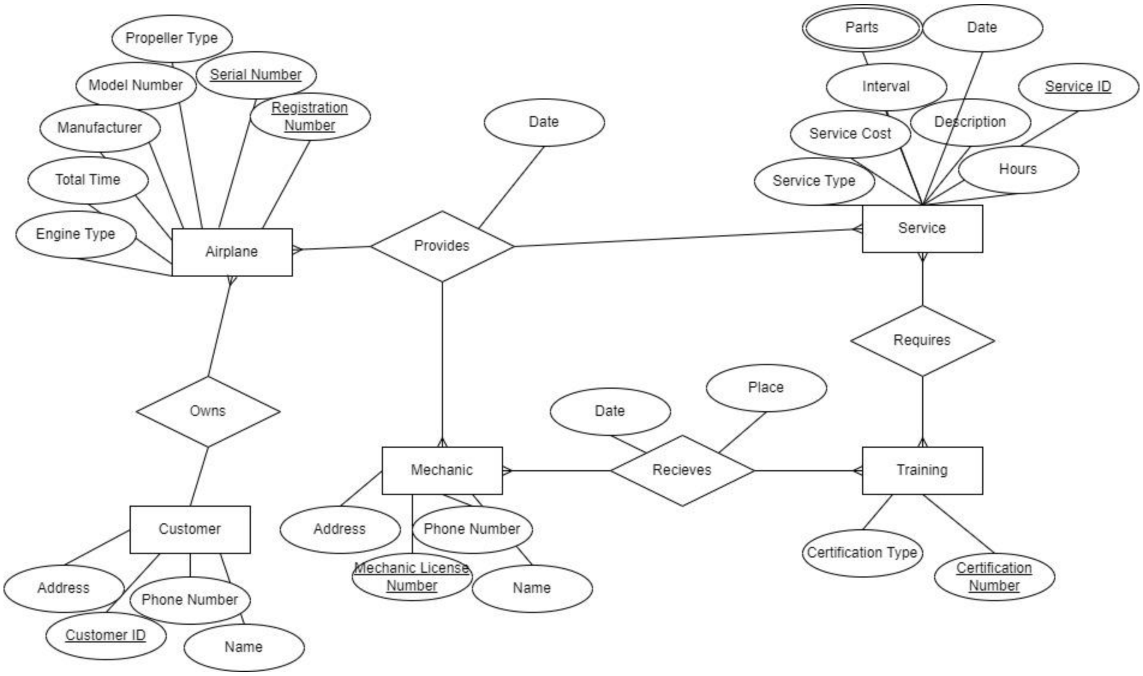


Figure 1: ER Diagram for Airplane Maintenance System with no Business Rule

A service requires many type of trainings, mechanic receives many type of trainings, and a mechanic provides many types of services for many types of airplanes. However, the question is: has the mechanic received the required training for the service he provides before providing that service? If not, he/she should not be allowed to provide such a service. This database model has no way of enforcing such a rule. If this is what we want, we better include it in our ER diagram (conceptual model) as a reminder for the programmer to implement it in his implementation which is not guaranteed that the programmer will consider it. To do so, some constraints are included in this ER model to enforce this type of business rule as can be seen in Figure 2 which will be implemented by writing trigger(s).

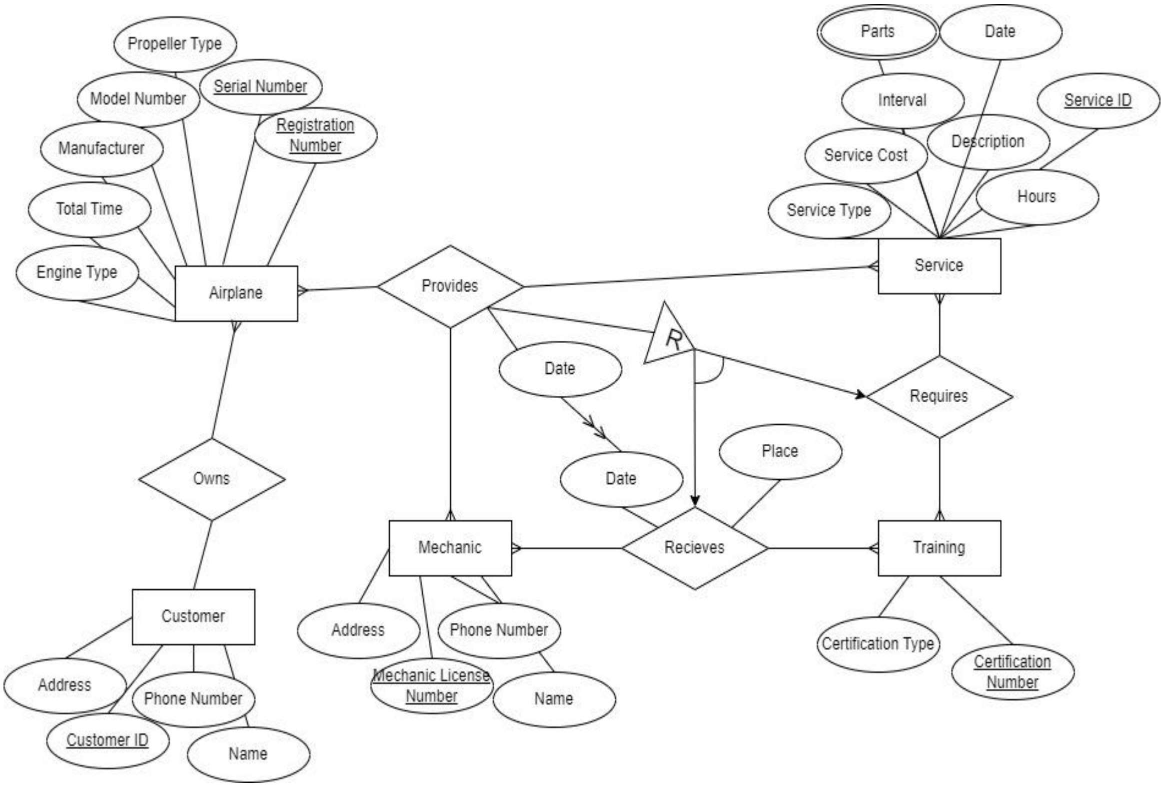


Figure 2: ER Diagram for Airplane Maintenance System Including Business Rules

A mechanic is an individual with certain skills that make him qualified to maintain airplanes. If the plane is serviced by an unqualified mechanic, the plane will not be certified to fly. So, the mechanic must receive specific types of training related to maintaining airplanes. There are many different types of training that a mechanic can receive for maintaining airplanes such as training on landing gear, training on engines, training on electronics, and so on. In turn the types of training that a mechanic receives are used to determine the types of maintenance services that the mechanic can perform on an airplane. A specific maintenance service may require that a mechanic receive more than one type of training. Yet, a specific type of training may be useful in providing more than a single maintenance service. To provide a service to an airplane, the mechanic must have received all the required training before the service is provided. Therefore, the date of providing service must be greater than the date of receiving the required training(s). Complex rules (constraints) like this can be implemented by writing triggers.

Provides table is a ternary relationship and becomes a table with four attributes, Serial-Number, Service-Id, Mechanic-License-Number, and data. Entering a row into this table means providing a service to a plane by a mechanic at a certain date. To make sure that the mechanic has received the required type of training for the type of service he is providing, the trigger needs to check the table Required to see what training are required for this service and then check the Receives table to make sure that this mechanic has received all trainings plus all the dates of receiving trainings are before the date of service before allowing this insertion into the Provides table.

Such a system guarantees that every service will be done by a qualified mechanic who has received all the required training for that service before the service is provided. Such a rule can’t be ignored because it will be implemented by writing a trigger and the trigger will be fired automatically.

The following sections represent relational implementation of this database model. Figure 3 is the schema for the ER diagram in Figure 2.

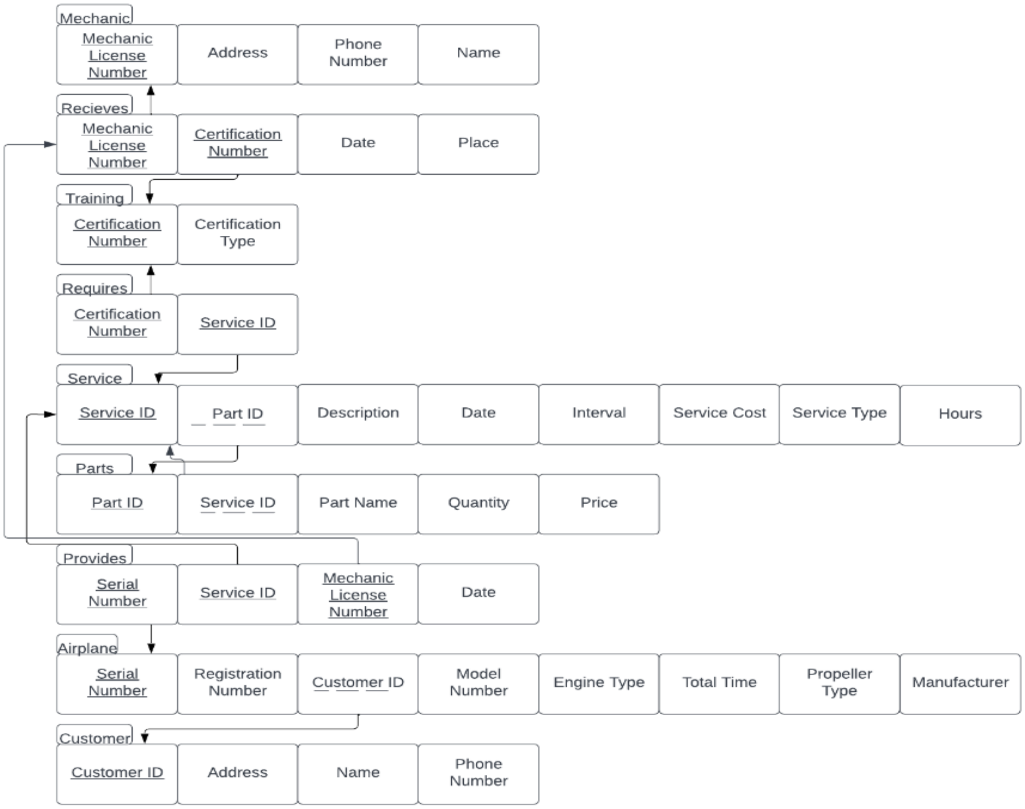


Figure 3: Schema for the ER Diagram

Here are SQL queries to create the tables and insert data into the tables using Oracle Database Management System. Some of these data are accessed from Diamond Aircraft website[3].

Mechanic Table:

```
CREATE TABLE  MECHANIC
   (        MECHANIC_LICENSE_NUMBER CHAR(8),
            ADDRESS VARCHAR2(64),
            PHONE_NUMBER CHAR(12),
            NAME VARCHAR2(64),
             CONSTRAINT MECHANIC_LICENSE_NUMBER_PK PRIMARY KEY
(MECHANIC_LICENSE_NUMBER) ENABLE
   ) ;
```

```
INSERT INTO Mechanic (Mechanic_License_Number, Address, Phone_Number, Name)
WITH names as (
SELECT 82345672, '123 Main St', '123-456-7890', 'Sarah Johnson' FROM dual UNION ALL
SELECT 10935645, '456 Park Ave', '234-567-8901', 'Michael Brown' FROM dual UNION ALL
SELECT 47952318, '789 Elm St', '345-678-9012', 'Jessica Davis' FROM dual UNION ALL
SELECT 59247638, '246 Oak Rd', '456-789-0123', 'David Wilson' FROM dual UNION ALL
SELECT 63750829, '135 Maple St', '567-890-1234', 'Karen Martinez' FROM dual
)
SELECT * FROM names
```

Training Table:

```
CREATE TABLE TRAINING
   (         CERTIFICATION_NUMBER CHAR(8),
             CERTIFICATION_TYPE VARCHAR2(64),
              CONSTRAINT TRAINING_PK PRIMARY KEY (CERTIFICATION_NUMBER)
ENABLE
   ) ;
```

```
INSERT INTO Training(Certification_Number, Certification_Type)
WITH names as (
SELECT 34918567, 'Mechanic License w/Airframe' FROM dual UNION ALL
SELECT 10239568, 'Mechanic License w/Powerplant' FROM dual UNION ALL
SELECT 75986032, 'Air Agency' FROM dual UNION ALL
SELECT 61453798, 'Inspection Authorization' FROM dual UNION ALL
SELECT 86725039, 'Air Agency' FROM dual
)
SELECT * FROM names
```

Receives Table:

```
CREATE TABLE RECIEVES
   (         MECHANIC_LICENSE_NUMBER CHAR(8),
             CERTIFICATION_NUMBER CHAR(8),
             DATE_RECIEVED DATE,
             PLACE VARCHAR2(64),
              CONSTRAINT RECIEVES_PK PRIMARY KEY (MECHANIC_LICENSE_NUMBER,
CERTIFICATION_NUMBER) ENABLE
   ) ;
```

```
INSERT INTO Recieves(Mechanic_License_Number, Certification_Number, Date_Recieved, Place)
WITH names as (
SELECT '82345672', '34918567', '01/30/2012', 'FAA' FROM dual UNION ALL
SELECT '10935645', '10239568', '06/15/2000', 'FAA' FROM dual UNION ALL
SELECT '47952318', '75986032', '12/01/2020', 'FAA' FROM dual UNION ALL
SELECT '59247638', '61453798', '08/22/2006', 'FAA' FROM dual UNION ALL
SELECT '63750829', '86725039', '04/04/2004', 'FAA' FROM dual
)
SELECT * FROM names
```

Service Table:

```
CREATE TABLE SERVICE
  (         SERVICE_ID CHAR(8),
            PART_ID CHAR(8),
            DESCRIPTION VARCHAR2(64),
            DATE_COMPLETED DATE,
            INTERVAL VARCHAR2(32),
            SERVICE_COST NUMBER,
            SERVICE_TYPE VARCHAR2(32),
            HOUR NUMBER,
             CONSTRAINT SERVICE_PK PRIMARY KEY (SERVICE_ID) ENABLE
  ) ;

INSERT INTO Service(Service_ID, Part_ID, Description, Date_Completed, Interval, Service_Cost,
Service_Type, Hour)
WITH names as (
SELECT '56231478', NULL, 'Inspect landing gear', '01/30/2020', 50, 430, 'Inspection', 8  FROM dual
UNION ALL
SELECT '52179568', '93245168', 'Oil change', '08/25/2022', NULL, 110, 'Maintenance', 2 FROM
dual UNION ALL
SELECT '98563215', '83124657', 'Replace and torque spark plugs', '04/02/2021', NULL, 160,
'Maintenance', 2 FROM dual UNION ALL
SELECT '12368460', NULL, 'Inspect cabin and cockpit', '03/25/2022', 50, 70, 'Inspection', 1 FROM
dual UNION ALL
SELECT '97256414', NULL, 'Inspect fuselage', '11/11/2021', 50, 130, 'Inspection', 1 FROM dual
)
SELECT * FROM names;
```

Parts Table:

```
CREATE TABLE PARTS
  (         PART_ID CHAR(8),
            SERVICE_ID CHAR(8),
            PART_NAME VARCHAR2(64),
            QUANTITY NUMBER,
            PRICE NUMBER,
             CONSTRAINT PARTS_PK PRIMARY KEY (PART_ID) ENABLE
  ) ;


INSERT INTO Parts(Part_ID, Service_ID, Part_Name, Quantity, Price)
WITH names as (
SELECT '93245168', '52179568', 'Oil', 1, 40 FROM dual UNION ALL
SELECT '83124657', '98563215', 'Spark plug', 4, 60 FROM dual
)
SELECT * FROM names;
```

Customer Table:

```
CREATE TABLE CUSTOMER
  (         CUSTOMER_ID CHAR(8),
            ADDRESS VARCHAR2(128),
            NAME VARCHAR2(64),
            PHONE_NUMBER VARCHAR2(12),
             CONSTRAINT Customer_pk PRIMARY KEY (CUSTOMER_ID) ENABLE
  ) ;

INSERT INTO Customer(Customer_ID, Address, Name, Phone_Number)
WITH names as (
SELECT '54123610', '116 E 1200S Springfield UT 64278', 'John Smith', '435-262-9857' FROM dual
UNION ALL
SELECT '25493582', '3456 Market St CA 94111', 'Thomas Matthews', '435-956-1246' FROM dual
UNION ALL
SELECT '72165893', '12 Oak Ave CA 90001', 'George Stephens', '801-659-8424' FROM dual
UNION ALL
SELECT '13489267', '567 Park Ave IL 60601', 'Mark Phillips', '234-562-4258' FROM dual UNION
ALL
SELECT '78214569', '100 Main St NY 10001', 'Jeff Lance', '216-654-2345' FROM dual
)
SELECT * FROM names;
```

Airplane Table:

```
CREATE TABLE  AIRPLANE
  (         AIRPLANE_SERIAL_NUMBER CHAR(8),
            REGISTRATION_NUMBER CHAR(6),
            CUSTOMER_ID CHAR(8),
            MODEL_NUMBER VARCHAR2(12),
            ENGINE_TYPE VARCHAR2(32),
            TOTAL_TIME NUMBER,
            PROPELLER_TYPE VARCHAR2(32),
            MANUFACTURER VARCHAR2(32),
             CONSTRAINT AIRPLANE_PK PRIMARY KEY (AIRPLANE_SERIAL_NUMBER)
ENABLE
  ) ;


INSERT INTO Airplane(Airplane_Serial_Number, Registration_Number, Customer_ID,
Model_Number, Engine_Type, Total_Time, Propeller_Type, Manufacturer)
WITH names as (
SELECT '13246589', 'N32465', '54123610', 'DV20 katana', 'Rotax 912ULS', 7821, 'Sensenich',
'Diamond Aircraft' FROM dual UNION ALL
SELECT '26489753', 'N62495', '25493582', 'DA20-A1', 'Jabiru 3300', 5367, 'XOAR', 'Diamond
Aircraft' FROM dual UNION ALL
SELECT '65498732', 'N32059', '72165893', 'DA50C', 'Corvair', 6325, 'Hoffmann', 'Diamond
Aircraft' FROM dual UNION ALL
SELECT '35126984', 'N03482', '13489267', 'HK36', 'HKS 700E TTS/TTC', 8624, 'Harzell', 'Diamond
Aircraft' FROM dual UNION ALL
SELECT '36247591', 'N63048', '78214569', 'DA62', 'Subaru EA81', 5634, 'MT-Propeller', 'Diamond
Aircraft' FROM dual
)
SELECT * FROM names;
```

Requires Table:

```
CREATE TABLE  REQUIRES
  (         CERTIFICATION_NUMBER CHAR(8),
            SERVICE_ID CHAR(8),
            CONSTRAINT REQUIRES_PK PRIMARY KEY (CERTIFICATION_NUMBER,
SERVICE_ID) ENABLE
   ) ;

INSERT INTO Requires(Certification_Number, Service_ID)
WITH names as (
SELECT '34918567', '56231478' FROM dual UNION ALL
SELECT '10239568', '52179568' FROM dual UNION ALL
SELECT '75986032', '98563215' FROM dual UNION ALL
SELECT '61453798', '12368460' FROM dual UNION ALL
SELECT '86725039', '97256414' FROM dual
)
SELECT * FROM names;
```

Provides Table:

```
CREATE TABLE PROVIDES
   (        Airplane_Serial_Number CHAR(8),
            SERVICE_ID CHAR(8),
            MECHANIC_LICENSE_NUMBER CHAR(8),
            DATE_PROVIDED DATE,
             CONSTRAINT PROVIDES_PK PRIMARY KEY (Airplane_Serial_Number,
SERVICE_ID, MECHANIC_LICENSE_NUMBER) ENABLE
   ) ;

ALTER TABLE  PARTS ADD CONSTRAINT PARTS_SERVICE_FK FOREIGN KEY
(SERVICE_ID)
          REFERENCES  SERVICE (SERVICE_ID) ENABLE;

ALTER TABLE  SERVICE ADD CONSTRAINT SERVICE_PARTS_FK FOREIGN KEY
(PART_ID)
          REFERENCES  PARTS (PART_ID) ENABLE;

ALTER TABLE  AIRPLANE ADD CONSTRAINT Airplane_Customer_fk FOREIGN KEY
(CUSTOMER_ID)
          REFERENCES  CUSTOMER (CUSTOMER_ID) ENABLE;
```

And here is the trigger to prevent providing service if the mechanic has not received all the required training before doing that service:

```
create or replace trigger PROVIDES_T1
BEFORE insert or update of Date_Provided on PROVIDES
FOR EACH ROW
declare
Date_Training_Recieved date;
Date_Of_Service date := :NEW.Date_Provided;
begin
SELECT Date_Recieved INTO Date_Training_Recieved FROM Recieves
WHERE Recieves.Mechanic_License_Number =
:NEW.Mechanic_License_Number;
```

```
if Date_Of_Service < Date_Training_Recieved THEN
   RAISE_APPLICATION_ERROR (-20201, 'Mechanic must recieve training
before he/she provides this service.' ||CHR(10)|| 'Date service is being provided:
'||Date_Of_Service || CHR(10)||'Date training has been recieved:
'||Date_Training_Recieved);
end if;
end;
```

The following queries are to insert rows into provides table which means providing services:

```
INSERT INTO Provides
VALUES('13246589', '56231478', '10935645', '05/21/2020')

INSERT INTO Provides
VALUES('26489753', '52179568', '10935645', '01/01/2016' )

INSERT INTO Provides
VALUES('65498732', '98563215', '59247638', '08/24/2018' )

INSERT INTO Provides
VALUES('65498732', '52179568', '63750829', '03/16/2019')

INSERT INTO Provides
VALUES('35126984', '98563215', '63750829', '12/27/2022')
```

The above insertions into Provides table work well because the mechanic has received the required training. The trigger allows the insertions.

However, the following insertions will not work because the mechanic has not received the required trainings. The trigger throws an error message and ignores the insertions.

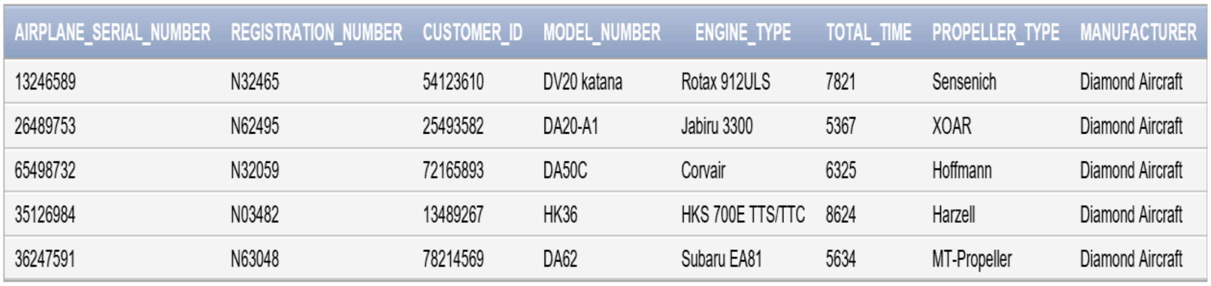
```
INSERT INTO Provides
VALUES('35126984', '98563215', '82345672', '12/27/2000')
```

```
ORA-20201: Mechanic must receive training before he/she provides this service.
Date service is being provided: 12/27/2000
Date training has been received: 01/30/2012
```

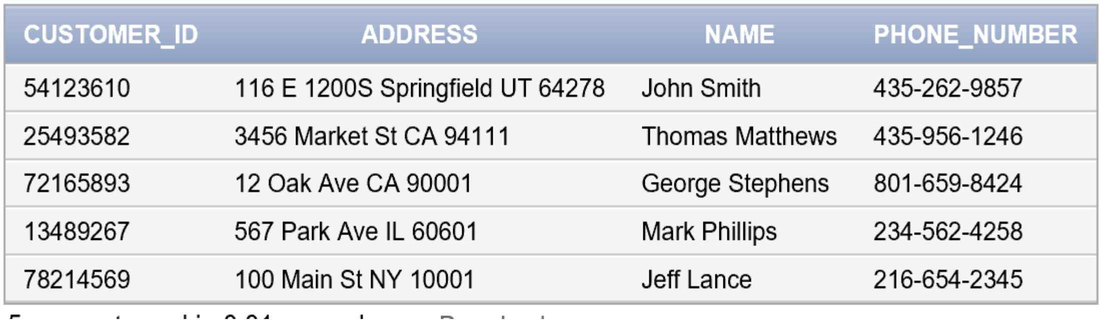
```
INSERT INTO Provides
VALUES('35126984', '98563215', '59247638', '07/27/2006')
```

```
ORA-20201: Mechanic must receive training before he/she provides this service.
Date service is being provided: 07/27/2006
Date training has been received 08/22/2006
```

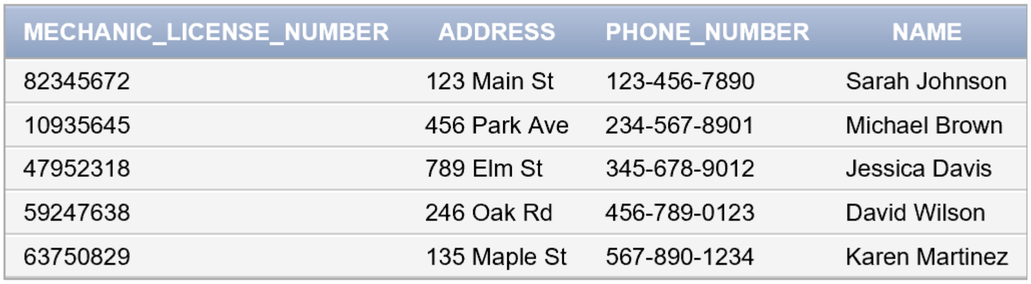
The following section shows the contents of these table after all the insertions:



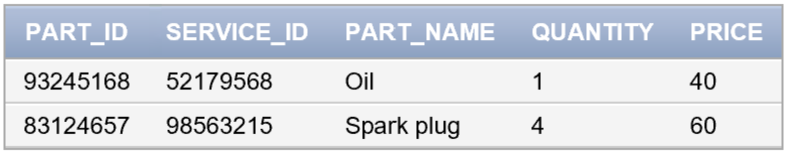
AIRPLANE_SERIAL_NUMBER	REGISTRATION_NUMBER	CUSTOMER_ID	MODEL_NUMBER	ENGINE_TYPE	TOTAL_TIME	PROPELLER_TYPE	MANUFACTURER
13246589	N32465	54123610	DV20 katana	Rotax 912ULS	7821	Sensenich	Diamond Aircraft
26489753	N62495	25493582	DA20-A1	Jabiru 3300	5367	XOAR	Diamond Aircraft
65498732	N32059	72165893	DA50C	Corvair	6325	Hoffmann	Diamond Aircraft
35126984	N03482	13489267	HK36	HKS 700E TTS/TTC	8624	Harzell	Diamond Aircraft
36247591	N63048	78214569	DA62	Subaru EA81	5634	MT-Propeller	Diamond Aircraft



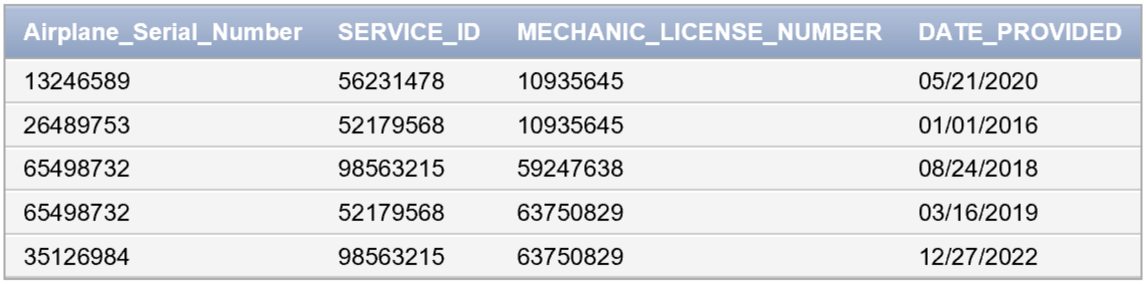
CUSTOMER_ID	ADDRESS	NAME	PHONE_NUMBER
54123610	116 E 1200S Springfield UT 64278	John Smith	435-262-9857
25493582	3456 Market St CA 94111	Thomas Matthews	435-956-1246
72165893	12 Oak Ave CA 90001	George Stephens	801-659-8424
13489267	567 Park Ave IL 60601	Mark Phillips	234-562-4258
78214569	100 Main St NY 10001	Jeff Lance	216-654-2345



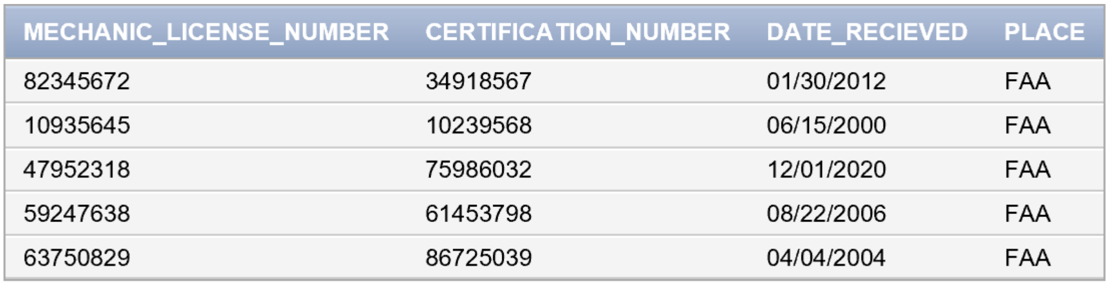
MECHANIC_LICENSE_NUMBER	ADDRESS	PHONE_NUMBER	NAME
82345672	123 Main St	123-456-7890	Sarah Johnson
10935645	456 Park Ave	234-567-8901	Michael Brown
47952318	789 Elm St	345-678-9012	Jessica Davis
59247638	246 Oak Rd	456-789-0123	David Wilson
63750829	135 Maple St	567-890-1234	Karen Martinez



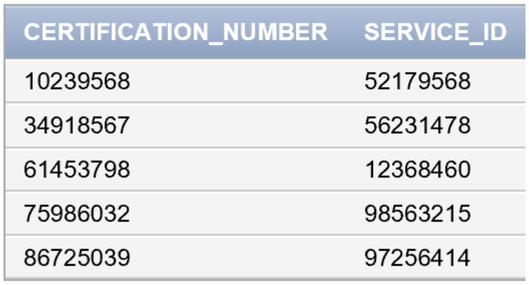
PART_ID	SERVICE_ID	PART_NAME	QUANTITY	PRICE
93245168	52179568	Oil	1	40
83124657	98563215	Spark plug	4	60



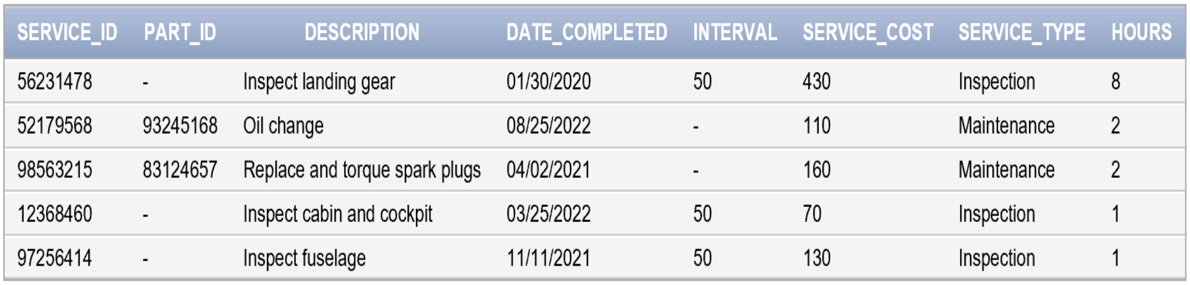
Airplane_Serial_Number	SERVICE_ID	MECHANIC_LICENSE_NUMBER	DATE_PROVIDED
13246589	56231478	10935645	05/21/2020
26489753	52179568	10935645	01/01/2016
65498732	98563215	59247638	08/24/2018
65498732	52179568	63750829	03/16/2019
35126984	98563215	63750829	12/27/2022



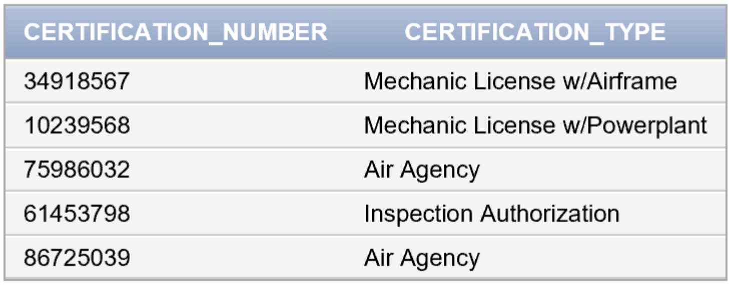
MECHANIC_LICENSE_NUMBER	CERTIFICATION_NUMBER	DATE_RECIEVED	PLACE
82345672	34918567	01/30/2012	FAA
10935645	10239568	06/15/2000	FAA
47952318	75986032	12/01/2020	FAA
59247638	61453798	08/22/2006	FAA
63750829	86725039	04/04/2004	FAA



CERTIFICATION_NUMBER	SERVICE_ID
10239568	52179568
34918567	56231478
61453798	12368460
75986032	98563215
86725039	97256414



SERVICE_ID	PART_ID	DESCRIPTION	DATE_COMPLETED	INTERVAL	SERVICE_COST	SERVICE_TYPE	HOURS
56231478	-	Inspect landing gear	01/30/2020	50	430	Inspection	8
52179568	93245168	Oil change	08/25/2022	-	110	Maintenance	2
98563215	83124657	Replace and torque spark plugs	04/02/2021	-	160	Maintenance	2
12368460	-	Inspect cabin and cockpit	03/25/2022	50	70	Inspection	1
97256414	-	Inspect fuselage	11/11/2021	50	130	Inspection	1



CERTIFICATION_NUMBER	CERTIFICATION_TYPE
34918567	Mechanic License w/Airframe
10239568	Mechanic License w/Powerplant
75986032	Air Agency
61453798	Inspection Authorization
86725039	Air Agency

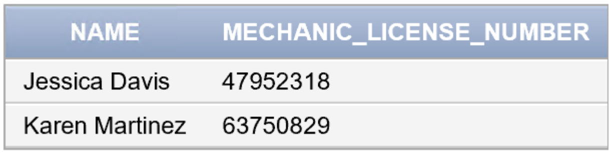
Here are some queries to retrieve information from these tables:

Query One:

Get the name and Id of every Mechanic with an Air Agency License Number

SELECT mechanic.name, mechanic.Mechanic_License_Number

```
FROM Mechanic
JOIN Recieves
ON Mechanic.Mechanic_License_Number = Recieves.Mechanic_License_Number
JOIN Training
ON Recieves.Certification_Number = Training.Certification_Number
WHERE Training.Certification_Type = 'Air Agency'
```

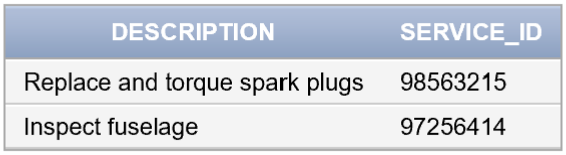


NAME	MECHANIC_LICENSE_NUMBER
Jessica Davis	47952318
Karen Martinez	63750829

Query Two:

Get the service descriptions and IDs that require an Air Agency certification

```
SELECT service.Description, service.Service_ID
FROM Service
JOIN Requires
ON Service.service_ID = Requires.service_ID
JOIN Training
ON Requires.certification_number = Training.Certification_Number
WHERE Certification_Type = 'Air Agency';
```

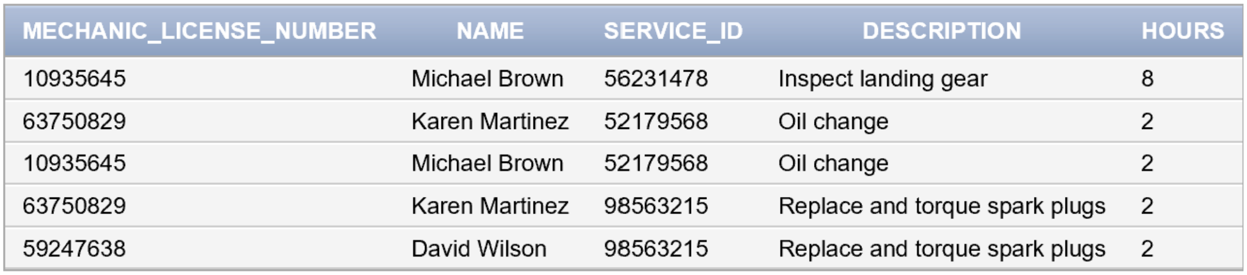


DESCRIPTION	SERVICE_ID
Replace and torque spark plugs	98563215
Inspect fuselage	97256414

Query Three:

Get the name, and mechanic license of each mechanic and the services they provided and the hours they spent on each service

```
SELECT Mechanic.Mechanic_License_Number, Mechanic.Name, Service.Service_ID,
Service.Description, service.Hours
FROM Mechanic
JOIN Provides
ON Mechanic.Mechanic_License_Number = Provides.Mechanic_License_Number
JOIN Service
ON Service.Service_Id = Provides.Service_ID
```



MECHANIC_LICENSE_NUMBER	NAME	SERVICE_ID	DESCRIPTION	HOURS
10935645	Michael Brown	56231478	Inspect landing gear	8
63750829	Karen Martinez	52179568	Oil change	2
10935645	Michael Brown	52179568	Oil change	2
63750829	Karen Martinez	98563215	Replace and torque spark plugs	2
59247638	David Wilson	98563215	Replace and torque spark plugs	2

Conclusion

In this paper, we have used the ER notation to represent some business rules graphically at conceptual level in a relational data model to enforce database integrity and/or consistency. The constraints that have been represented in this paper make sure that the service to airplane is provided by a qualified mechanic. Continuing with our airplane servicing system we learn that each type of service requires specific tools. Also, airplane service takes place in certain locations such as hanger where the tolls are kept. The future of this work will include additional constraints to ensure the correctness of operations such as "a maintenance service is only provided in a hanger using proper tolls by a well-qualified mechanic". Also when a service to an airplane is done, a stored procedure is executed to print an invoice for the customer.

This is a sample project that students in our database class implement to get hands-on experience.

References:

[1] J. A. Hoffer, M. B. Prescott and F. R. McFadden, "*Modern Database Management*", Seventh Edition, Prentice Hall, 2005.

[2] *A. Perkins, "Business Rules = Meta Data*", the proceedings of the: Technology of Object-Oriented Languages and Systems, IEEE, 2000.

[3] http://support.diamond-air.at/techpubs+M52087573ab0.html:

[4] Reza Sanati Mehrizy, Curtis Welborn, and Afsaneh Minaie, "*Representing and Enforcing Business Rules in Relational Data Model",* American Society for Engineering Education (ASEE) Annual Conference, June 2006.