12/11/2017

Rennehan,Richard

NSCC dATABASE ADMinistration

Assignment 3 Oracle SQL

DBAS4041

# Introduction

The following report is a collection of SQL statements needed to create the database for Tiny College.

Note that the SQL statements are made for Oracle SQL Developer. Tweaking may be needed if another database will be used

Legend:

|  |  |
| --- | --- |
|  |  |
| TERMS |  |
| NOT NULL | Field cannot be left blank |
| PRIMARY KEY | Field cannot be left blank and must be unique. No repeats. |
| FOREIGN KEY | Field has a connection to a primary key field in another table. Changes made in one table may affect related data in another |
| INNER JOIN | Allows tables to be linked together to form a single table |
| /\* \*/ | Used for commenting inside code. Whatever is inside does not affect the code |
|  |  |
| ID RANGES |  |
| 10000- | Departments |
| 20000- | Parts |
| 30000- | Vehicles |
| 40000- | Faculty Members |
| 50000- | Mechanics |
| 60000- | TFBS Members |
| 70000- | Reservations |
| 80000- | Maintenance Reports |

# SQL Create Table Statements

## Department Table

/\*Creates table for departments\*/

CREATE TABLE DEPARTMENT (

DEP\_ID INTEGER,

DEP\_NAME VARCHAR(80) NOT NULL,

DEP\_AREACODE CHAR(3) NOT NULL,

DEP\_PHONE CHAR(8) NOT NULL,

CONSTRAINT dep\_pk PRIMARY KEY(DEP\_ID)

);

## Vehicle Table

/\*Creates vehicle table for storing vehicle stock and information\*/

CREATE TABLE VEHICLE (

VEH\_ID INTEGER,

VEH\_TYPE VARCHAR(32) NOT NULL,

VEH\_MODEL VARCHAR(48) NOT NULL,

VEH\_MILEAGERATE NUMBER NOT NULL,

VEH\_CURRENTODO NUMBER NOT NULL,

CONSTRAINT veh\_pk PRIMARY KEY(VEH\_ID)

);

## Part Table

/\*Creates part table for storing part information and stock\*/

CREATE TABLE PART (

PART\_ID INTEGER,

PART\_NAME VARCHAR(64) NOT NULL,

PART\_PRICE NUMBER NOT NULL,

PART\_QOH NUMBER NOT NULL,

PART\_MINIMUM INTEGER NOT NULL,

CONSTRAINT part\_pk PRIMARY KEY(PART\_ID)

);

## Mechanic Table

/\*Creates table for mechanics\*/

CREATE TABLE MECHANIC (

MECH\_ID INTEGER,

MECH\_LNAME VARCHAR(32) NOT NULL,

MECH\_FNAME VARCHAR(32) NOT NULL,

MECH\_AREACODE CHAR(3) NOT NULL,

MECH\_PHONE CHAR(8) NOT NULL,

MECH\_ADDRESS VARCHAR(64) NOT NULL,

MECH\_AUTHORIZED CHAR(1) NOT NULL,

CONSTRAINT mech\_pk PRIMARY KEY(MECH\_ID)

);

## TFBS\_Member Table

/\*Creates table for TFBS members\*/

CREATE TABLE TFBS\_MEMBER (

TFBS\_ID INTEGER,

TFBS\_LNAME VARCHAR(32) NOT NULL,

TFBS\_FNAME VARCHAR(32) NOT NULL,

TFBS\_AREACODE CHAR(3) NOT NULL,

TFBS\_PHONE CHAR(8) NOT NULL,

TFBS\_ADDRESS VARCHAR(64) NOT NULL,

CONSTRAINT tfbs\_pk PRIMARY KEY(TFBS\_ID)

);

## Faculty\_Member Table

/\*Creates table for faculty members\*/

CREATE TABLE FACULTY\_MEMBER (

FAC\_ID INTEGER,

FAC\_LNAME VARCHAR(32) NOT NULL,

FAC\_FNAME VARCHAR(32) NOT NULL,

DEP\_ID INTEGER NOT NULL,

VEH\_ID INTEGER,

FAC\_AREACODE CHAR(3) NOT NULL,

FAC\_PHONE CHAR(8) NOT NULL,

FAC\_ADDRESS VARCHAR(64) NOT NULL,

CONSTRAINT fac\_pk PRIMARY KEY(FAC\_ID),

CONSTRAINT fac\_dep\_fk FOREIGN KEY(DEP\_ID) REFERENCES DEPARTMENT(DEP\_ID)

ON DELETE CASCADE,

CONSTRAINT fac\_veh\_fk FOREIGN KEY(VEH\_ID) REFERENCES VEHICLE(VEH\_ID)

ON DELETE CASCADE

);

## Reservation Table

/\*Creates reservation table for storing trip information\*/

CREATE TABLE RESERVATION (

RES\_ID INTEGER NOT NULL,

DEP\_ID INTEGER NOT NULL,

RES\_DEPARTINGDATE DATE NOT NULL,

RES\_FINISHDATE DATE NOT NULL,

RES\_DESTINATION VARCHAR(80) NOT NULL,

FAC\_ID INTEGER NOT NULL,

VEH\_ID INTEGER NOT NULL,

RES\_ODOSTART NUMBER NOT NULL,

RES\_ODOEND NUMBER NOT NULL,

RES\_MAINTCOMPLAINTS VARCHAR(200),

RES\_FUELPURCHASED NUMBER,

RES\_CREDITCARD CHAR(16),

CONSTRAINT res\_pk PRIMARY KEY(RES\_ID),

CONSTRAINT res\_fac\_fk FOREIGN KEY(FAC\_ID) REFERENCES FACULTY\_MEMBER(FAC\_ID)

ON DELETE CASCADE,

CONSTRAINT res\_veh\_fk FOREIGN KEY(VEH\_ID) REFERENCES VEHICLE(VEH\_ID)

ON DELETE CASCADE,

CONSTRAINT res\_dep\_fk FOREIGN KEY(DEP\_ID) REFERENCES DEPARTMENT(DEP\_ID)

ON DELETE CASCADE

);

## Maintenance Table

/\*Creates maintenance table\*/

CREATE TABLE MAINTENANCE (

MAINT\_ID INTEGER,

MECH\_ID INTEGER NOT NULL,

VEH\_ID INTEGER NOT NULL,

MAINT\_INSEPCTTYPE VARCHAR(160) NOT NULL,

MAINT\_ENTRYDATE DATE NOT NULL,

MAINT\_COMPLDATE DATE NOT NULL,

RES\_ID INTEGER,

CONSTRAINT maint\_pk PRIMARY KEY(MAINT\_ID),

CONSTRAINT maint\_mech\_fk FOREIGN KEY(MECH\_ID) REFERENCES MECHANIC(MECH\_ID)

ON DELETE CASCADE,

CONSTRAINT maint\_veh\_fk FOREIGN KEY(VEH\_ID) REFERENCES VEHICLE(VEH\_ID)

ON DELETE CASCADE,

CONSTRAINT maint\_res\_fk FOREIGN KEY(RES\_ID) REFERENCES RESERVATION(RES\_ID)

ON DELETE CASCADE

);

## Maintenance\_Detail Table

/\*Creates maintenance detail table for recording additional maintenance actions\*/

CREATE TABLE MAINTENANCE\_DETAIL (

MAINT\_ID INTEGER NOT NULL,

MAINTD\_LINENUM NUMBER NOT NULL,

PART\_ID INTEGER,

MAINTD\_AMOUNTUSED NUMBER ,

MECH\_ID INTEGER NOT NULL,

CONSTRAINT maintd\_maint\_fk FOREIGN KEY(MAINT\_ID) REFERENCES MAINTENANCE(MAINT\_ID)

ON DELETE CASCADE,

CONSTRAINT maintd\_mech\_fk FOREIGN KEY(MECH\_ID) REFERENCES MECHANIC(MECH\_ID)

ON DELETE CASCADE,

CONSTRAINT maintd\_part\_fk FOREIGN KEY(PART\_ID) REFERENCES PART(PART\_ID)

ON DELETE CASCADE

);

# SQL Insert into Statements

## Insert Departments into Database

/\*Inserts the departments into the system\*/

INSERT ALL

 INTO DEPARTMENT VALUES (10000, 'Rennehan Auto', '902', '555-1000')

 INTO DEPARTMENT VALUES (10001, 'Rons Depot', '902', '555-1001')

SELECT \* FROM DUAL/\*Dummy line to allow the command to function\*/;

## Insert Vehicles into Database

/\*Inserts the vehicles into the system\*/

INSERT ALL

 INTO VEHICLE VALUES (30000, 'Sedan', 'Toyota Yaris iA', 0.40, 75000)

 INTO VEHICLE VALUES (30001, 'Van', 'Dodge Grand Caravan', 0.55, 100000)

SELECT \* FROM DUAL/\*Dummy line to allow the command to function\*/;

## Insert Parts into Database

/\*Inserts the parts inventory into the system\*/

INSERT ALL

 INTO PART VALUES (20000, 'Belt', 50, 30, 4)

 INTO PART VALUES (20001, 'Tire', 150, 50, 8)

 INTO PART VALUES (20002, 'Battery', 75, 10, 2)

SELECT \* FROM DUAL/\*Dummy line to allow the command to function\*/;

## Insert People into Database

/\*Inserts all employees into the system\*/

INSERT ALL

 INTO FACULTY\_MEMBER VALUES (40000, 'McLeod', 'Ron', 10001, NULL,

 '902', '555-2001', '2223 Base St.')

INTO FACULTY\_MEMBER VALUES (40001, 'Gillespie', 'Geoff', 10001, 30000,

'902', '555-2002', '1481 Prog St.')

 INTO FACULTY\_MEMBER VALUES (40002, 'Rennehan', 'Richard', 10000, 30001,

 '902', '555-2003', '5685 Leeds St.')

 INTO FACULTY\_MEMBER VALUES (40003, 'Bradfield', 'Jordan', 10000, 30001,

 '902', '555-2004', '5685 Leeds St.')

 INTO MECHANIC VALUES (50000, 'Doe', 'Jane', '902', '555-3000', '32 Fixerupper St.', 'y')

 INTO MECHANIC VALUES (50001, 'Doe', 'John', '902', '555-3001', '32 Fixerupper St.', 'y')

 INTO TFBS\_MEMBER VALUES (60000, 'Hoffman', 'Mark', '902', '555-4000', '20 Water St.')

 INTO TFBS\_MEMBER VALUES (60001, 'Davis', 'Andrew', '902', '555-4001', '33 Main Ave.')

SELECT \* FROM DUAL/\*Dummy line to allow the command to function\*/;

## Insert Reservation Data into Database

/\*Inserts example reservations into the system\*/

INSERT ALL

 INTO RESERVATION VALUES (70000, 10001, '12-DEC-17', '19-DEC-17', 'Learning Centre', 40000,30000, 74500, 75000, NULL, NULL, NULL)

 INTO RESERVATION VALUES (70001, 10000, '14-DEC-17', '21-DEC-17', 'Research Facility', 40001, 30001, 99750, 100000, 'Squeaking noise when starting up and accelerating', NULL, NULL)

 INTO RESERVATION VALUES (70002, 10000, '27-DEC-17', '29-DEC-17', 'Learning Centre', 40003, 30000, 100000, 100500, 'Vehicle excessively vibrates', NULL, NULL)

SELECT \* FROM DUAL/\*Dummy line to allow the command to function\*/;

## Insert Maintenance Data into Database

/\*Inserts example maintenance data into the system\*/

INSERT ALL

 INTO MAINTENANCE VALUES (80000, 50001, 30001, 'Under the hood', '22-DEC-17',

'23-DEC-17',70001)

 INTO MAINTENANCE\_DETAIL VALUES (80000, 1, 20000, 1, 50001)

 INTO MAINTENANCE VALUES (80001, 50000, 30000, 'Test drive', '30-DEC-17', '31-DEC-17', 70002)

 INTO MAINTENANCE\_DETAIL VALUES (80001, 1, 20000, 1, 50000)

 INTO MAINTENANCE\_DETAIL VALUES (80001, 2, 20001, 2, 50000)

SELECT \* FROM DUAL/\*Dummy line to allow the command to function\*/;

# SQL Reports

## Parts Usage Report

/\*Amount used per part\*/

SELECT PART.PART\_ID, PART.PART\_NAME, SUM(MAINTD\_AMOUNTUSED) AS AMOUNT\_USED

FROM (PART INNER JOIN MAINTENANCE\_DETAIL ON PART.PART\_ID = MAINTENANCE\_DETAIL.PART\_ID)

INNER JOIN MAINTENANCE ON MAINTENANCE\_DETAIL.MAINT\_ID = MAINTENANCE.MAINT\_ID

WHERE MAINT\_COMPLDATE BETWEEN '01-DEC-17' AND '31-DEC-17'

GROUP BY PART.PART\_ID, PART.PART\_NAME

ORDER BY SUM(MAINTD\_AMOUNTUSED);

## Vehicle Mileage Report

/\*Miles driven per vehicle\*/

SELECT VEHICLE.VEH\_ID, VEH\_TYPE, VEH\_MODEL, SUM(RES\_ODOEND - RES\_ODOSTART) AS DISTANCE

FROM RESERVATION INNER JOIN VEHICLE ON RESERVATION.VEH\_ID = VEHICLE.VEH\_ID

WHERE RES\_FINISHDATE BETWEEN '01-DEC-17' AND '31-DEC-17'

GROUP BY VEHICLE.VEH\_ID, VEH\_TYPE, VEH\_MODEL

ORDER BY VEH\_TYPE;

## Department Mileage Report

/\*Miles driven per department\*/

SELECT RESERVATION.DEP\_ID, DEP\_NAME, SUM(RES\_ODOEND - RES\_ODOSTART) AS DISTANCE

FROM RESERVATION INNER JOIN DEPARTMENT ON RESERVATION.DEP\_ID = DEPARTMENT.DEP\_ID

WHERE RES\_FINISHDATE BETWEEN '01-DEC-17' AND '31-DEC-17'

GROUP BY RESERVATION.DEP\_ID, DEP\_NAME

ORDER BY SUM(RES\_ODOEND - RES\_ODOSTART) DESC;

## Vehicle Revenue Report

/\*Profits earned per vehicle\*/

SELECT VEHICLE.VEH\_ID, VEH\_TYPE, VEH\_MODEL, VEH\_MILEAGERATE, SUM(VEH\_MILEAGERATE\*(RES\_ODOEND - RES\_ODOSTART)) AS REVENUE

FROM VEHICLE INNER JOIN RESERVATION ON VEHICLE.VEH\_ID = RESERVATION.VEH\_ID

WHERE RES\_FINISHDATE BETWEEN '01-DEC-17' AND '31-DEC-17'

GROUP BY VEHICLE.VEH\_ID, VEH\_TYPE, VEH\_MODEL, VEH\_MILEAGERATE

ORDER BY VEH\_TYPE;

## Department Revenue Report

/\*Profits earned per department\*/

SELECT DEPARTMENT.DEP\_ID, DEP\_NAME, SUM((RES\_ODOEND - RES\_ODOSTART)\*VEH\_MILEAGERATE) AS REVENUE

FROM (DEPARTMENT INNER JOIN RESERVATION ON DEPARTMENT.DEP\_ID = RESERVATION.DEP\_ID)

INNER JOIN VEHICLE ON RESERVATION.VEH\_ID = VEHICLE.VEH\_ID

WHERE RES\_FINISHDATE BETWEEN '01-DEC-17' AND '31-DEC-17'

GROUP BY DEPARTMENT.DEP\_ID, DEP\_NAME

ORDER BY DEP\_NAME;