INFO20003 – Database Systems

Semester 2 2016

Assignment 2

Ivan Ken Weng Chee

736901

Question 1

```
/* Assumed that AUTO_INCREMENT does not change the specification,
 * and NO ACTION is done after DELETE and UPDATE commands
*/
-- Table `Location`
DROP TABLE IF EXISTS `Location`;

CREATE TABLE IF NOT EXISTS `Location` (
        `idLocation` INT NOT NULL AUTO_INCREMENT,
        `StreetNumber` SMALLINT NULL,
        `StreetNumberSuffix` VARCHAR(20) NULL,
        `StreetName` VARCHAR(50) NULL,
        `StreetType` VARCHAR(20) NULL,
        `MinorMunicipality` VARCHAR(50) NULL,
        `MajorMunicipality` VARCHAR(50) NULL,
        `GoverningDistrict` VARCHAR(50) NULL,
        `PostalArea` VARCHAR(4) NULL,
        `Country` VARCHAR(50) NULL,
        PRIMARY KEY (`idLocation`)
) ENGINE = InnoDB;

-- Table `Consumer`
DROP TABLE IF EXISTS `Consumer`;

CREATE TABLE IF NOT EXISTS `Consumer` (
        `UserName` VARCHAR(50) NOT NULL,
        `FirstName` VARCHAR(45) NULL,
        `LastName` VARCHAR(45) NULL,
        `email` VARCHAR(100) NULL,
        PRIMARY KEY (`UserName`)
) ENGINE = InnoDB;

-- Table `ConsumerLocation`
DROP TABLE IF EXISTS `ConsumerLocation`;

CREATE TABLE IF NOT EXISTS `ConsumerLocation` (
        `UserName` VARCHAR(50) NOT NULL,
        `idLocation` INT NOT NULL,
        `ValidFrom` DATE NOT NULL,
        `ValidTo` DATE NULL,
        PRIMARY KEY (`UserName`, `idLocation`, `ValidFrom`),
        INDEX `fk_ConsumerLocation_Consumer1_idx` (`UserName` ASC),
        INDEX `fk_ConsumerLocation_Location1_idx` (`idLocation` ASC),
        CONSTRAINT `fk_ConsumerLocation_Consumer1`
                FOREIGN KEY (`UserName`)
        REFERENCES `Consumer` (`UserName`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
        CONSTRAINT `fk_ConsumerLocation_Location1`
                FOREIGN KEY (`idLocation`)
        REFERENCES `Location` (`idLocation`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
) ENGINE = InnoDB;

-- Table `ConsumerLocation`
DROP TABLE IF EXISTS `Licence`;
```

```
CREATE TABLE IF NOT EXISTS `Licence` (
    `SoftwareId` INT NOT NULL AUTO_INCREMENT,
    `UserName` VARCHAR(50) NOT NULL,
    `PurchaseTime` DATETIME NOT NULL,
    `Installed` BOOLEAN NOT NULL,
    PRIMARY KEY (`SoftwareId` , `UserName` , `PurchaseTime`),
    INDEX `fk_Licence_Software1_idx` (`SoftwareId` ASC),
    INDEX `fk_Licence_Consumer1_idx` (`UserName` ASC),
    CONSTRAINT `fk_Licence_Software1` FOREIGN KEY (`SoftwareId`)
        REFERENCES `Software` (`SoftwareId`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_Licence_Consumer1` FOREIGN KEY (`UserName`)
        REFERENCES `Consumer` (`UserName`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)  ENGINE = InnoDB;
```

Question 2

```
SET @PID = 0;
SET @SID = 0;

-- Consumer 1
INSERT IGNORE INTO Consumer VALUES (
        "ichee",
        "Ivan Ken Weng",
        "Chee",
        "ichee@student.unimelb.edu.au"
);

-- Location 1
SET @CID = "ichee";
INSERT IGNORE INTO Location VALUES (
        DEFAULT,
        1,
        "Some Suffix",
        "Some",
        "Street",
        "Some Minor Municipality",
        "Some Major Municipality",
        "Some Governing District",
        "1234",
        "Batmania"
);

-- ConsumerLocation 1
SET @LID = LAST_INSERT_ID();
INSERT IGNORE INTO ConsumerLocation VALUES (
        @CID,
        @LID,
        "2016-09-28",
        NULL
);

-- Platform 1 - iOS
SET @PID = @PID + 1;
INSERT IGNORE INTO Platform VALUES (
        @PID,
        "iOS"
);

-- Software 1 - Gougle Docs
SET @SID = @SID + 1;
INSERT IGNORE INTO Software VALUES (
        @SID,
```

```
        "Gougle Docs",
        1,
        1000.00,
        250.00,
        @PID,
        "Professional software collection to edit and view various document formats",
        2016,
        "http://docs.gougle.com"
);

-- Licence 1
INSERT IGNORE INTO Licence VALUES (
        @SID,
        @CID,
        "2016-09-28",
        TRUE
);

-- Consumer 2
INSERT IGNORE INTO Consumer VALUES (
        "batman",
        "Bruce",
        "Wayne",
        "thedarknight@dccomics.com"
);

-- Location 2
SET @CID = "batman";
INSERT IGNORE INTO Location VALUES (
        DEFAULT,
        1007,
        "Wayne Manor",
        "Mountain",
        "Drive",
        "Wayne Enterprises",
        "Gotham City",
        "New Jersey",
        "0000",
        "United States"
);

-- ConsumerLocation 2
SET @LID = LAST_INSERT_ID();
INSERT IGNORE INTO ConsumerLocation VALUES (
        @CID,
        @LID,
        "2016-09-29",
        NULL
);

-- Platform 2 - OS X
SET @PID = @PID + 1;
INSERT IGNORE INTO Platform VALUES (
        @PID,
        "OS X"
);

-- Software 2 - Gougle Music
SET @SID = @SID + 1;
INSERT IGNORE INTO Software VALUES (
        @SID,
        "Gougle Music",
        10,
        45.00,
        10.00,
        @PID,
        "Synchronise your music, videos and podcasts to all your Gougle devices",
```

```
        2015,
        "http://music.gougle.com"
);

-- Licence 2
INSERT IGNORE INTO Licence VALUES (
        @SID,
        @CID,
        "2016-09-29",
        FALSE
);

-- Consumer 3
INSERT IGNORE INTO Consumer VALUES (
        "harry",
        "Harry",
        "Potter",
        "harry@hogwarts.com"
);

-- Location 3
SET @CID = "harry";
INSERT IGNORE INTO Location VALUES (
        DEFAULT,
        4,
        "Stair Cupboard",
        "Privet",
        "Drive",
        "Little Whinging",
        "Surrey",
        "London",
        "1511",
        "England"
);

-- ConsumerLocation 3
SET @LID = LAST_INSERT_ID();
INSERT IGNORE INTO ConsumerLocation VALUES (
        @CID,
        @LID,
        "2016-09-30",
        NULL
);

-- Platform 3 - Windows
SET @PID = @PID + 1;
INSERT IGNORE INTO Platform VALUES (
        @PID,
        "Windows"
);

-- Software 3 - Gougle Chrome
SET @SID = @SID + 1;
INSERT IGNORE INTO Software VALUES (
        @SID,
        "Gougle Chrome",
        6,
        0.00,
        10.00,
        @PID,
        "Fastest way to browse the web. Download using Microhard Internet Explorer 11",
        2010,
        "http://www.chrome.com"
);

-- Licence 3
INSERT IGNORE INTO Licence VALUES (
```

```
        @SID,
        @CID,
        "2016-09-30",
        TRUE
);

-- Consumer 4
INSERT IGNORE INTO Consumer VALUES (
        "ironman",
        "Tony",
        "Stark",
        "ironman@avengers.com"
);

-- Location 4
SET @CID = "ironman";
INSERT IGNORE INTO Location VALUES (
        DEFAULT,
        10880,
        "Stark Malibu Mansion",
        "Malibu",
        "Point",
        "Point Dume",
        "Malibu",
        "California",
        "9026",
        "United States"
);

-- ConsumerLocation 4
SET @LID = LAST_INSERT_ID();
INSERT IGNORE INTO ConsumerLocation VALUES (
        @CID,
        @LID,
        "2016-10-01",
        NULL
);

-- Platform 4 - Android
SET @PID = @PID + 1;
INSERT IGNORE INTO Platform VALUES (
        @PID,
        "Android"
);

-- Software 4 - Gougle Drive
SET @SID = @SID + 1;
INSERT IGNORE INTO Software VALUES (
        @SID,
        "Gougle Drive",
        3,
        25.00,
        15.00,
        @PID,
        "Access all your data on the cloud, anytime, anywhere",
        2014,
        "http://drive.gougle.com"
);

-- Licence 4
INSERT IGNORE INTO Licence VALUES (
        @SID,
        @CID,
        "2016-10-01",
        FALSE
);
```

```
-- Consumer 5
INSERT IGNORE INTO Consumer VALUES (
        "spiderman",
        "Peter",
        "Parker",
        "spiderman@notavengers.com"
);

-- Location 5
SET @CID = "spiderman";
INSERT IGNORE INTO Location VALUES (
        DEFAULT,
        20,
        "Aunt May's Home",
        "Ingram",
        "Street",
        "Forest Hills Gardens",
        "Flushing",
        "New York",
        "1375",
        "United States"
);

-- ConsumerLocation 5
SET @LID = LAST_INSERT_ID();
INSERT IGNORE INTO ConsumerLocation VALUES (
        @CID,
        @LID,
        "2016-10-02",
        NULL
);

-- Platform 5 - Playstation
SET @PID = @PID + 1;
INSERT IGNORE INTO Platform VALUES (
        @PID,
        "PlayStation"
);

-- Software 5 - Gougle PlayStore
SET @SID = @SID + 1;
INSERT IGNORE INTO Software VALUES (
        @SID,
        "Gougle PlayStore",
        4,
        0.00,
        3.00,
        @PID,
        "Download apps and games from the largest repository of mobile apps",
        2011,
        "http://play.gougle.com"
);

-- Licence 5
INSERT IGNORE INTO Licence VALUES (
        @SID,
        @CID,
        "2016-10-02",
        FALSE
);
```

Question 3

```
/* Assumed that old location instances are kept in the database,
 * even when no current consumers are bounded to them
 */
-- Creates a new Location instance
INSERT IGNORE INTO Location Values (
```

```
        DEFAULT,
        123,
        "",
        "Fake",
        "Street",
        "Fakeville",
        "Melbourne",
        "Victoria",
        "9999",
        "Australia"
);

SET @LID = LAST_INSERT_ID();
SET @DID = "2016-07-10";
SET @CID = (
        SELECT Consumer.UserName
        FROM Consumer
        WHERE Consumer.FirstName = "Ivan Ken Weng"
        AND Consumer.LastName = "Chee"
);

-- Updates ValidTo date of pre-existing Location
UPDATE ConsumerLocation
SET ConsumerLocation.ValidTo = @DID
WHERE ConsumerLocation.UserName = @CID;

-- Creates a new ConsumerLocation instance to
-- link the new location instance to myself
INSERT INTO ConsumerLocation VALUES (
        @CID,
        @LID,
        @DID,
        NULL
);
```

Question 4

```
/* Includes software where both software price and distribution cost is zero
 */
SELECT COUNT(DISTINCT Software.SoftwareId) AS NumApps
FROM Software INNER JOIN Platform
ON Software.idPlatform = Platform.idPlatform
WHERE Platform.Name = "iOS"
AND Software.DistributionCost < (0.2 * Software.Price)
```

Question 5

```
/* Assumed that a decade starts on a year ending with '0' and ends ending with '9'
 */
SELECT Decades.Decade, COUNT(DISTINCT Software.SoftwareId) AS iOSApps
FROM Software INNER JOIN (
        SELECT Software.SoftwareId, FLOOR(Software.YearOfRelease / 10) * 10 AS Decade
        FROM Software INNER JOIN Platform
        ON Software.idPlatform = Platform.idPlatform
        WHERE Platform.Name = "iOS"
        GROUP BY Software.SoftwareId
        HAVING Decade < 2010
) Decades
ON Decades.SoftwareId = Software.SoftwareId
GROUP BY Decades.Decade
```

Question 6

```
SELECT Developers.idStaff,
       Developers.OfficialJobTitle,
       Developers.FirstName,
       Developers.LastName,
       COUNT(iOS.SoftwareId) AS NumiOSApps
FROM (
```

```
        SELECT DISTINCT Development.idStaff,
                        JobTitle.OfficialJobTitle,
                        Staff.FirstName,
                        Staff.LastName,
                        Software.SoftwareId
        FROM Development INNER JOIN Staff INNER JOIN Software INNER JOIN JobTitle
        ON Development.idStaff = Staff.idStaff
        AND Development.idJobTitle = JobTitle.idJobTitle
        AND Development.SoftwareId = Software.SoftwareId
        WHERE JobTitle.OfficialJobTitle = "Software Developer"
        GROUP BY Development.idStaff, Software.SoftwareId
) Developers LEFT JOIN (
        SELECT Software.SoftwareId
        FROM Development INNER JOIN Software INNER JOIN Platform
        ON Development.SoftwareId = Software.SoftwareId
        AND Software.idPlatform = Platform.idPlatform
        WHERE Platform.Name = "iOS"
) iOS
ON Developers.SoftwareId = iOS.SoftwareId
GROUP BY Developers.idStaff
```

Question 7

```
/* Query returns two developers by default - most active and least active,
 * unless there is a tie for max and min number of apps developed
 * Assumed two developers will be returned even if both most active
 * and least active developers are the same person
 */
SELECT Development.idStaff,
       Staff.FirstName,
       Staff.LastName,
       COUNT(Development.idStaff) AS NumProjects
FROM Staff INNER JOIN Development
ON Staff.idStaff = Development.idStaff
GROUP BY Development.idStaff
HAVING COUNT(Development.idStaff) = (
        SELECT MAX(Developers.NumProjects) AS MaxProjects
        FROM Staff INNER JOIN (
                SELECT Development.idStaff, COUNT(Development.idStaff) AS NumProjects
                FROM Staff INNER JOIN Development
                ON Staff.idStaff = Development.idStaff
                GROUP BY Development.idStaff
        ) Developers
        ON Staff.idStaff = Developers.idStaff
)
OR COUNT(Development.idStaff) = (
        SELECT MIN(Developers.NumProjects) AS MinProjects
        FROM Staff INNER JOIN (
                SELECT Development.idStaff, COUNT(Development.idStaff) AS NumProjects
                FROM Staff INNER JOIN Development
                ON Staff.idStaff = Development.idStaff
                GROUP BY Development.idStaff
        ) Developers
        ON Staff.idStaff = Developers.idStaff
)
```

Question 8

```
/* Assumed purchased software are those which are associated to a licence,
 * and have price greater than $0.00
 */
SELECT DISTINCT COUNT(Licence.PurchaseTime) AS TimesPurchased,
                Software.SoftwareId,
                Software.Name,
                Software.Price,
                Software.Description,
                Software.Website
FROM Licence INNER JOIN Software
ON Licence.SoftwareId = Software.SoftwareId
```

```
WHERE Software.Price > 0
GROUP BY Licence.SoftwareId, Software.Name
HAVING BINARY(Software.Name) REGEXP '^[i]'
OR BINARY(Software.Name) REGEXP '^[e]'
ORDER BY TimesPurchased DESC
LIMIT 10;
```

Question 9

```
/* Assumed software that has not been purchased includes those with price = $0.00
 * (free software), and that domain name is the part of the web address before '/'
 */
SELECT DISTINCT Software.Name,
SUBSTRING_INDEX(SUBSTRING_INDEX(Software.Website, 'http://', - 1), '/', 1) AS Domain
FROM Software
WHERE Software.SoftwareId NOT IN (
        Select DISTINCT Licence.SoftwareId
        FROM Licence
)
ORDER BY Software.Name
```

Question 10

```
/* Assumed that a decade starts on a year ending with '0' and ends ending with '9'
*/
SELECT COUNT(DISTINCT Software.Name) AS SoftwareInstalled
FROM Software INNER JOIN Licence INNER JOIN Consumer INNER JOIN ConsumerLocation INNER
JOIN Location
ON Software.SoftwareId = Licence.SoftwareId
AND Licence.UserName = Consumer.UserName
AND Consumer.UserName = ConsumerLocation.UserName
AND ConsumerLocation.idLocation = Location.idLocation
WHERE Software.CurrentVersion = 1
AND Licence.Installed = TRUE
AND Location.Country = "Australia"
ORDER BY Software.Name
```

Question 11

```
/* Assumed that paying consumers have bought software with price > $0.00,
 * and location popularity is based on number of occupants, instead of
 * number of paying consumers/total amount paid by consumers
 */

SELECT Paying.idLocation,
       Occupants.NumOccupants,
       Location.StreetNumber,
       Location.StreetNumberSuffix,
       Location.StreetName,
       Location.StreetType,
       Location.MinorMunicipality,
       Location.MajorMunicipality,
       Location.GoverningDistrict,
       Location.PostalArea,
       Location.Country
FROM (
        SELECT DISTINCT Location.idLocation
        FROM Location INNER JOIN ConsumerLocation INNER JOIN Licence INNER JOIN
Software
        ON Location.idLocation = ConsumerLocation.idLocation
        AND ConsumerLocation.UserName = Licence.UserName
        AND Licence.SoftwareId = Software.SoftwareId
        WHERE Software.Price > 0
        GROUP BY Location.idLocation
) Paying INNER JOIN (
        SELECT COUNT(ConsumerLocation.idLocation) AS NumOccupants,
        ConsumerLocation.idLocation
        FROM ConsumerLocation
        GROUP BY ConsumerLocation.idLocation
) Occupants INNER JOIN Location
```

```
ON Paying.idLocation = Occupants.idLocation
AND Paying.idLocation = Location.idLocation
ORDER BY NumOccupants DESC
LIMIT 10
```

Question 12

```
/* Assumed that consumers who never had a location recorded in the system
 * are those with no association with any ConsumerLocation entity
 */
SELECT DISTINCT Consumer.UserName, Consumer.FirstName, Consumer.LastName
FROM Consumer
WHERE Consumer.UserName NOT IN (
        SELECT DISTINCT ConsumerLocation.UserName
        FROM ConsumerLocation
)
```