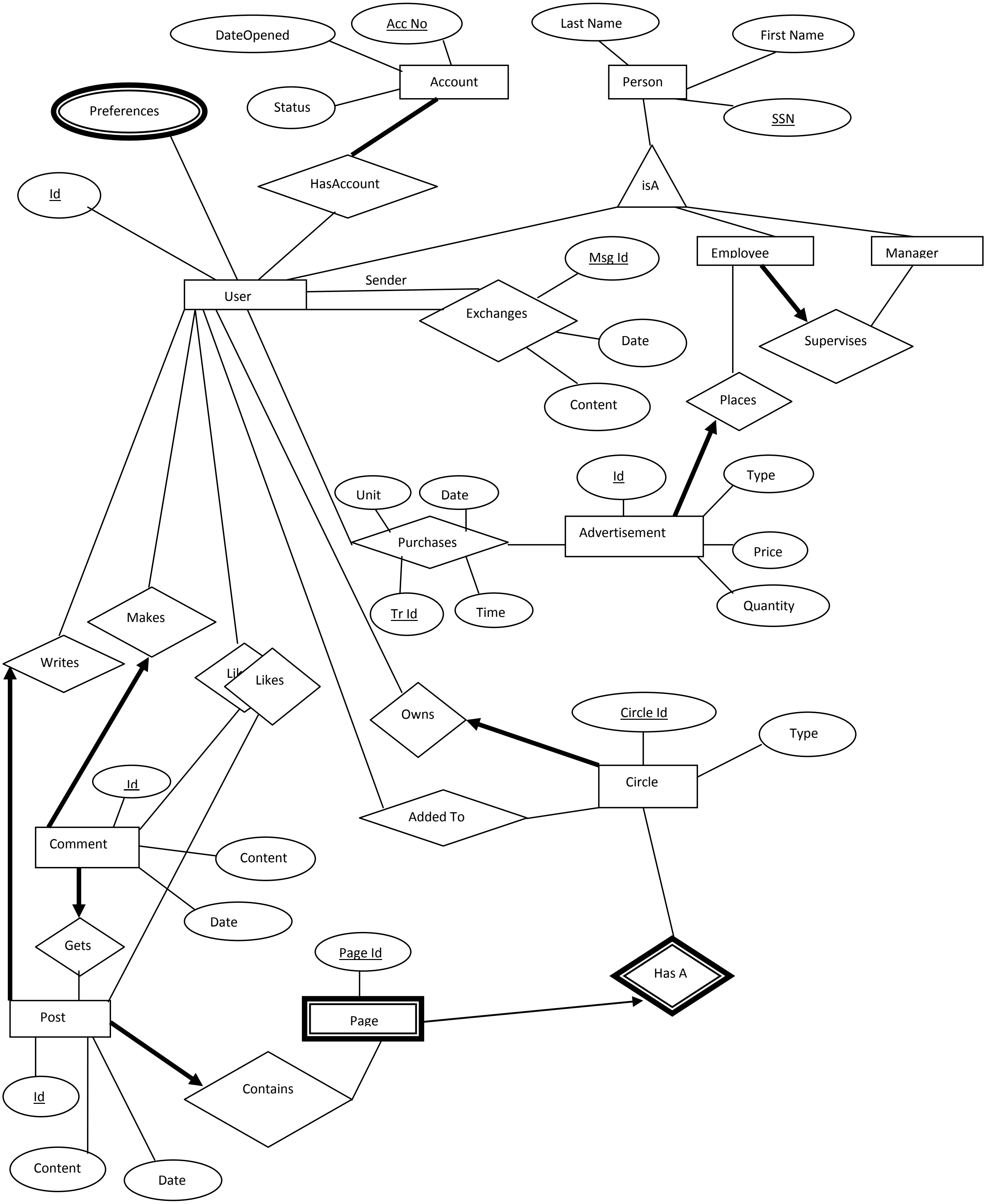
FB+ ER Diagram



Notes on the E-R Model:

1) Entity type Page is a *weak entity type* which is dependent on entity type Circle. Please see Section 4.4.3 of the course textbook for a discussion of weak entity types and the Part-of relationship.

2) The E-R model does *not* guarantee that a user will only be subjected to advertisements for which he has a preference. Further consideration of this constraint is deferred until Project Assignment No. 3.

3) The overlapping Likes relationship types indicate that there are actually two binary Likes relationships: one relating User to Post and the other relating User to Comment.

4) According to the project specification, the edge between User and Purchases should actually be between Account and Purchases. Making this change would result in some layout issues in the diagram, which we wanted to avoid at this time.

5) The tables defined below are not defined in any particular order, so simply attempting to copy-paste the CREATE TABLE statements into the SQL command editor will most likely throw exceptions. A better approach would be to define the tables in a well-defined order, or to add primary-key and other constraints separately using ALTER TABLE statements.

Relational Model

```
CREATE TABLE Circle(
Circle_Id                    INTEGER,
Circle_NAME                  VARCHAR(100),
Owner_Of_Circle              INTEGER,
Type                         VARCHAR(50),
PRIMARY KEY(Circle_Id),
FOREIGN KEY (Owner_Of_Circle) REFERENCES User(User_Id),
CHECK (Circle_Id > 0)
);

CREATE TABLE User(
SSN                          INTEGER,
User_Id                      INTEGER,
Account_Creation_Date        DATETIME NOT NULL ,
Rating                       INTEGER,
PRIMARY KEY(User_Id),
CHECK (User_Id>0),
FOREIGN KEY (SSN) REFERENCES Person(SSN)
);
```

```
CREATE TABLE AddedTo(
User_Id                           INTEGER,
Circle_Id                         INTEGER,
PRIMARY KEY(UserId,Circle_Id),
FOREIGN KEY (UserId) REFERENCES User(User_Id),
FOREIGN KEY (CircleId) REFERENCES Circle(Circle_Id)
);
CREATE TABLE Account(
Account_Number                    INTEGER,
Account_Creation_Date             DATETIME NOT NULL,
Credit_Card_Number                INTEGER,
Status                            VARCHAR(10),
PRIMARY KEY(Account_Number),
CHECK(Account_Number>0)
);

CREATE TABLE User_Has_Account(
User_Id                           INTEGER,
Account_Number                    INTEGER,
PRIMARY KEY (User_id, Account_Number),
FOREIGN KEY(Account_Number) REFERENCES Account(Account_Number),
FOREIGN KEY(User_Id) REFERENCES  User(User_Id)
);

CREATE TABLE User_Preferences(
Id                                INTEGER,
Preference                        VARCHAR(50),
PRIMARY KEY (Id, Preference),
FOREIGN KEY(Id) REFERENCES User(User_Id)
);

CREATE TABLE Message(
Message_Id                        INTEGER,
Date                              DATETIME NOT NULL ,
Subject                           VARCHAR(50),
Content                           VARCHAR(1000),
Sender                            INTEGER,
Receiver                          INTEGER,
PRIMARY KEY (Message_Id) ,
CHECK (Message_Id>0),
FOREIGN KEY (Sender) REFERENCES User(User_Id),
FOREIGN KEY (Receiver) REFERENCES User(User_Id)
);
```

```
CREATE TABLE Person(
SSN                             INTEGER,
Last_Name                       VARCHAR(50),
First_Name                      VARCHAR(50),
Address                         VARCHAR(100),
City                            VARCHAR(50),
State                           VARCHAR(50),
Zip_Code                        INTEGER,
Telephone                       INTEGER,
Email_Address                   VARCHAR(50),
PRIMARY KEY (SSN),
CHECK (SSN>0)
);

CREATE TABLE Employee(
SSN                             INTEGER,
Start_Date                      DATETIME NOT NULL,
Hourly_Rate                     INTEGER,
Manager                         INTEGER,
PRIMARY KEY (SSN),
FOREIGN KEY (Manager) REFERENCES Manager(SSN),
FOREIGN KEY (SSN) REFERENCES Person(SSN)
);

CREATE TABLE Manager(
SSN                             INTEGER,
Start_Date                      DATETIME NOT NULL,
Hourly_Rate                     INTEGER,
PRIMARY KEY (SSN),
FOREIGN KEY (SSN) REFERENCES Person(SSN)
);

CREATE TABLE Post(
Post_Id                         INTEGER,
Date                            DATETIME NOT NULL ,
Content                         VARCHAR(50),
Comment_Count                   INTEGER,
Circle                          INTEGER,
Author                          INTEGER,
PRIMARY KEY (Post_Id),
FOREIGN KEY (Circle) REFERENCES Circle(Circle_Id),
FOREIGN KEY (AUTHOR) REFERENCES User(User_Id),
CHECK (Post_Id>0 AND Comment_Count>=0)
);

CREATE TABLE Comment(
Comment_Id                      INTEGER,
Date                            DATETIME NOT NULL ,
Content                         VARCHAR(50),
Post                            INTEGER,
Author                          INTEGER,
PRIMARY KEY (Comment_Id),
CHECK (Comment_Id>0),
FOREIGN KEY (Post) REFERENCES Post(Post_Id),
FOREIGN KEY (AUTHOR) REFERENCES User(User_Id)
);

CREATE TABLE User_Likes_Post(
User                            INTEGER,
Post                            INTEGER,
PRIMARY KEY(User,Post),
FOREIGN KEY (User) REFERENCES User(User_Id),
FOREIGN KEY (Post) REFERENCES Post(Post_Id)
);

CREATE TABLE User_Likes_Comment(
User                            INTEGER,
Comment                         INTEGER,
PRIMARY KEY(User,Comment),
FOREIGN KEY (User) REFERENCES User(User_Id),
FOREIGN KEY (Comment) REFERENCES Comment(Comment_Id)
);

CREATE TABLE Advertisement(
Advertisement_Id                INTEGER,
Employee                        INTEGER,
Type                            VARCHAR(50),
Date                            DATETIME NOT NULL ,
Company                         VARCHAR(50),
Item_Name                       VARCHAR(50),
Content                         VARCHAR(50),
Unit_Price                      INTEGER,
Availabe_Units                  INTEGER,
PRIMARY KEY (Advertisement_Id),
CHECK (Advertisement_Id>0),
FOREIGN KEY (Employee) REFERENCES Employee(SSN)
);

CREATE TABLE Purchase(
Transaction_Id                  INTEGER,
Date                            DATETIME NOT NULL ,
Advertisement                   INTEGER,
Number_Of_Units                 INTEGER,
Account                         INTEGER,
User                            INTEGER,
PRIMARY KEY (Transaction_Id),
CHECK (TRANSACTION_Id>0),
FOREIGN KEY (Advertisement) REFERENCES Advertisement(Advertisement_Id),
FOREIGN KEY (User,Account) REFERENCES User_Has_Account(User_Id, Account_Number)
);
```