5.1)  
Rule 3: Attributes in a subtype do not match inherited attributes.  
• Subtype Entity 4 has Attribute 2-1 listed, even though it would receive this from its supertype, Entity 2. 

Rule 6: Weak entity types have at least one identifying relationship.  
• Weak Entity 1 has no identifying relationships. Either make the entity regular, or add an identifying relationship. 

Rule 7: For each identifying relationship, at least one participating entity type must be weak.  
• Rel 6 is an identifying relationship but neither of its entities (6 and 7) are weak. Either make the relationship not identifying, or make at least one of the entities weak. Cardinality may have to change in order to follow Rule 8 if Entity 7 is made weak. 

Rule 9: Redundant foreign keys are not used.  
• Entity 6 and 7 each have Attribute 7-1. Remove attribute from Entity 6.  
• Entity 3 and 6 each have Attribute 6-1. Remove attribute from Entity 3. 

Comments:  
2 points for each rule violation. Reduced 1 point if only 1 violation identified in rule 9 or if rule 3 violation is classified under rule 9 violation.
Comments:
1 point reduced if no relationship between physician->physician_appt or patient->appointment or person->appointment.
2 points reduced if relationships not named.
2 points for violating consistency rules.
3 points for not using crow’s foot notation.
6.1) Read problem 6.20 (page 199) and apply the information to fill in the tables with the matching blanks in the following ERD. The missing information could be entities, attributes or relationships. Additionally, fill in the table about cardinality.
Fill in the answers only in the table given below. Do not write your answers directly into the ER diagram.

<table>
<thead>
<tr>
<th>EntityID</th>
<th>Entity Name</th>
<th>Identifying attribute</th>
<th>Other attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Customer</td>
<td>1. CustID</td>
<td>2. CustName</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. CustStreet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4. CustCity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5. CustState</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6. CustZip</td>
</tr>
<tr>
<td>2.</td>
<td>Work Order</td>
<td>7. WorkNo</td>
<td>8. WorkCreationDate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>9. WorkDateRequire</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10. WorkCompletionDate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11. WorkCust</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>12. WorkSupervising</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13. WorkCity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>14. WorkZip</td>
</tr>
<tr>
<td>3.</td>
<td>16. Task</td>
<td>TaskID</td>
<td>TaskName</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>17. TaskEstimatedHrs</td>
</tr>
<tr>
<td>4.</td>
<td>Task_On_WorkOrder</td>
<td>Status Actual hours</td>
<td>18. CompletionDate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relationship ID</th>
<th>Relationship Name</th>
<th>Entity 1</th>
<th>Entity 2</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>15. Submits</td>
<td>Customer</td>
<td>Work Order</td>
<td>1-M</td>
</tr>
<tr>
<td>2.</td>
<td>19. Contains</td>
<td>Work Order</td>
<td>Task_on_Workload</td>
<td>1-M</td>
</tr>
<tr>
<td>3.</td>
<td>20. Assigned</td>
<td>Task_on_Workload</td>
<td>16. Task</td>
<td>1-M</td>
</tr>
</tbody>
</table>

*If you understood Task_on_Workload as Task in the table above, and answered with M-N relationship, no point is cut for it.

Revise your ERD to meet the additional requirements described in the last 3 bullets of page 199.
6.2) Answer conversion problem 6.4 from the textbook, page 213

Final result:
CREATE TABLE Employee
(
    EmpNo   INTEGER,
    EmpName   VARCHAR(50),
CONSTRAINT PKEmployee PRIMARY KEY (EmpNo)
)

CREATE TABLE Skill (
    SkillNo   INTEGER,
    SkillName VARCHAR(50),
CONSTRAINT PKSkill PRIMARY KEY (SkillNo)
)

CREATE TABLE Project (
    ProjNo   INTEGER,
    ProjName  VARCHAR(50),
CONSTRAINT PKSkill PRIMARY KEY (ProjNo)
)

CREATE TABLE Provides (
    EmpNo   INTEGER NOT NULL,
    SkillNo   INTEGER NOT NULL,
    ProjNo   INTEGER NOT NULL,
    Hrs   INTEGER NOT NULL,
CONSTRAINT PKProvides PRIMARY KEY (EmpNo, SkillNo, ProjNo),
CONSTRAINT FKEmpNo FOREIGN KEY (EmpNo) REFERENCES Employee,
CONSTRAINT FKSkillNo FOREIGN KEY (SkillNo) REFERENCES Skill,
CONSTRAINT FKProjNo FOREIGN KEY (ProjNo) REFERENCES Project
)
6.3) Answer conversion problem 6.10 from the textbook, page 216

Final result:

```sql
CREATE TABLE Provider
(
  ProvNo CHAR(11),
  ProvFirstName VARCHAR(50),
  ProvLastName VARCHAR(50),
  ProvPhone CHAR(7),
  ProvSpecialty VARCHAR(50),
CONSTRAINT PKProvider PRIMARY KEY (ProvNo)
)

CREATE TABLE Physician
(
  ProvNo CHAR(11),
  PhyEMail VARCHAR(30),
  PhyHospital VARCHAR(30),
  PhyCertification VARCHAR(20),
CONSTRAINT PKPhysician PRIMARY KEY (ProvNo),
CONSTRAINT FKProvNo FOREIGN KEY (ProvNo) REFERENCES Provider
ON DELETE CASCADE
)

CREATE TABLE Nurse
(
  ProvNo CHAR(11),
  NursePayGrade INTEGER,
  NurseTitle VARCHAR(20),
CONSTRAINT PKNurse PRIMARY KEY (ProvNo),
CONSTRAINT FKProvNo FOREIGN KEY (ProvNo) REFERENCES Provider
ON DELETE CASCADE
)

CREATE TABLE Patient
(
  PatNo CHAR(11),
  PatFirstName VARCHAR(15),
  PatLastName VARCHAR(15),
  PatCity VARCHAR(15),
  PatState CHAR(2),
  PatZip CHAR(5),
  PatHealthPlan VARCHAR(30),
CONSTRAINT PKPatient PRIMARY KEY (PatNo)
)

CREATE TABLE Visit
(
  VisitNo VARCHAR(11),
  VisitDate DATE,
  VisitPayMethod VARCHAR(30),
  VisitCharge DECIMAL(8,2),
  PatNo CHAR(11) NOT NULL,
  ProvNo CHAR(11) NOT NULL,
CONSTRAINT PKVisit PRIMARY KEY (VisitNo),
CONSTRAINT FKPatNo FOREIGN KEY (PatNo) REFERENCES Patient,
CONSTRAINT FKProvNo FOREIGN KEY (ProvNo) REFERENCES Physician
)
CREATE TABLE Item
( ItemNo CHAR(11),
  ItemDesc VARCHAR(60),
  ItemType VARCHAR(10),
  ItemPrice DECIMAL(7,2),
CONSTRAINT PKItem PRIMARY KEY (ItemNo))

CREATE TABLE VisitDetail
( VisitDetailNo CHAR(11),
  VisitNo CHAR(11),
  ItemNo CHAR(11) NOT NULL,
  DetailCharge DECIMAL(7,2),
CONSTRAINT PKVisitDetail PRIMARY KEY (VisitDetailNo, VisitNo),
CONSTRAINT FKVisitNo FOREIGN KEY (VisitNo) REFERENCES Visit,
CONSTRAINT FKItemNo FOREIGN KEY (ItemNo) REFERENCES Item)

CREATE TABLE Provides
( ProvNo CHAR(11),
  VisitDetailNo CHAR(11),
  VisitNo CHAR(11),
CONSTRAINT PKVisitDetail PRIMARY KEY (ProvNo, VisitDetailNo, VisitNo),
CONSTRAINT FKProvNo FOREIGN KEY (ProvNo) REFERENCES Nurse,
CONSTRAINT FKVisitDetailNo FOREIGN KEY (VisitDetailNo, VisitNo) REFERENCES VisitDetail)