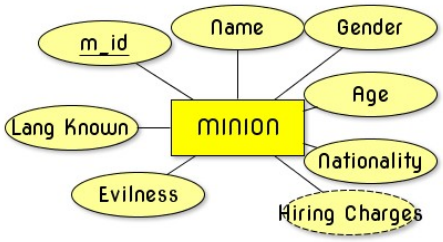
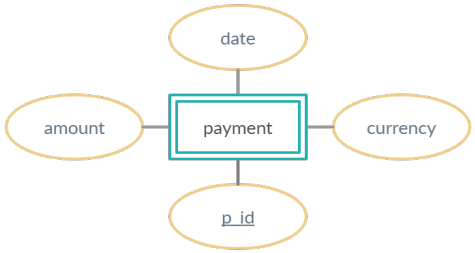
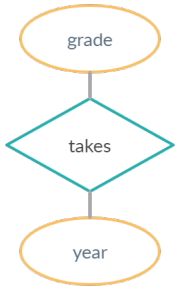
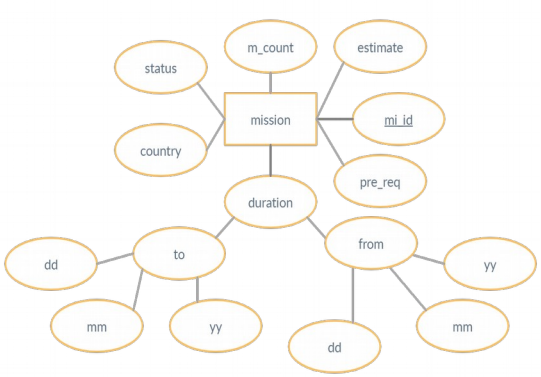
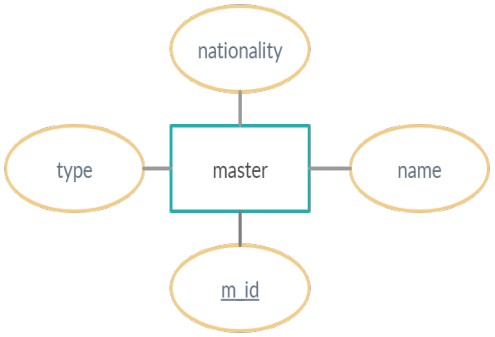
ER-RELATIONAL MAPPING

<table>
<tr><th>ER COMPONENT</th><th>TYPE/ PRIMARY KEY(S)</th><th colspan="2">RELATED ENTITIES & PRIMARY KEY</th><th>CORRESPONDING RELATION SCHEMA</th></tr>
<tr><td rowspan="2"> 
</td><td>Strong entity</td><td>Evilness</td><td>e_id</td><td rowspan="2">Create table minion(
m_id number(4),
name varchar(15),
gender char(1),
age number(1),
nationality varchar(25),
hiring charge number(10,3),
evilness number(2),
lang_known varchar(30)
);</td></tr>
<tr><td>m_id</td><td>Training</td><td>t_id</td></tr>
<tr><td rowspan="2"> 
</td><td>Weak entity</td><td>currency</td><td>c_id</td><td rowspan="2">Create table payment
(
mi_id number(4), p_id number(4),
currency varchar(10), amount
number(10,3), primary
key(mi_id,p_id), foreign key(mi_id)
references mission(mi_id)
on delete cascade
);</td></tr>
<tr><td>p_id</td><td>mission</td><td>mi_id</td></tr>
<tr><td rowspan="2"> 
</td><td>relation</td><td>training</td><td>t_id</td><td rowspan="2">Create table takes(
m_id number(4),
t_id number(4),
year number(4),
grade varchar(2),
primary key(m_id,t_id),
foreign key (m_id)
references minion(m_id),
foreign key (t_id)
references training(t_id)
);</td></tr>
<tr><td>minion(m_id)

training(t_id)</td><td>minion</td><td>m_id</td></tr>
</table>

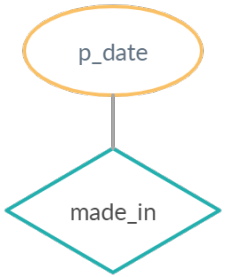
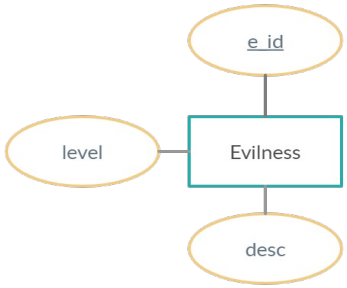
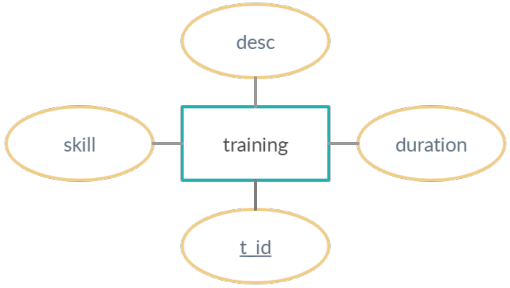
<table>
<tr><th>ER COMPONENT</th><th>TYPE/ PRIMARY KEY(S)</th><th colspan="2">RELATED ENTITIES & PRIMARY KEY</th><th>CORRESPONDING RELATION SCHEMA</th></tr>
<tr><td rowspan="3"> 
</td><td rowspan="3">Strong entity</td><td>minion</td><td>minion (m_id)</td><td rowspan="3">Create table mission
(mi_id number(4),
m_id number(4),
estimate number(10,3),
m_count number(2),
pre_req varchar(25),
status varchar(15),
country varchar(15),
from_date date,
to_date date,

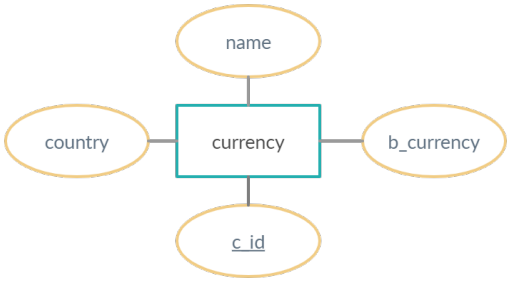
primary key(mi_id),
foreign key(m_id) references master(m_id),

);</td></tr>
<tr><td rowspan="2">mi_id</td><td>master</td><td>master (m_id)</td></tr>
<tr><td>payment</td><td>p_id</td></tr>
<tr><td rowspan="3"> 
</td><td rowspan="2">Strong entity</td><td>mission</td><td>mi_id</td><td rowspan="3">Create table master(
m_id number(4),
name varchar(15),
type varchar(15),

nationality varchar(25),
primary key(m_id,),

);</td></tr>
<tr><td rowspan="2">m_id</td><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>

ER COMPONENT	TYPE/ PRIMARY KEY(S)	RELATED ENTITIES & PRIMARY KEY		CORRESPONDING RELATION SCHEMA
	relation	payment	p_id	**Create table payment_made_in(p_id number(4), c_id number(4), p_date date, primary key(p_id,c_id,date), foreign key (p_id) references payment(p_id), foreign key (c_id) references currency(c_id));**
	p_date, payment(p_id), currency(c_id)	currency	c_id	
	Strong entity	minion	m_id	**Create table evilness (e_id number(4), level number(2), desc varchar(25), primary key(e_id),);**
	e_id	training	t_id	
	Strong entity	minion	m_id	**Create table training (t_id number(4), skill varchar(15), duration number(10,3), desc varchar(25), primary key(t_id),);**
	t_id	evilness	e_id	

ER COMPONENT	TYPE/ PRIMARY KEY(S)	RELATED ENTITIES & PRIMARY KEY		CORRESPONDING RELATION SCHEMA
	Strong entity	payment	p_id	**Create table currency(** **c_id number(4),** **name varchar(15),** **country varchar(25),** **b_currency number(10,3),** **primary key(c_id)** **);**
	c_id			
	relation	minion	m_id	**Create table has(** **m_id number(4),** **e_id number(4),** **primary key(m_id,e_id),** **foreign key (m_id)** **references mission(m_id),** **foreign key (e_id)** **references evilness(e_id)** **);**
	mission(m_id) **evilness(e_id)**	evilness	e_id	
	relation	evilness	e_id	**Create table provides(** **t_id number(4),** **e_id number(4),** **foreign key (t_id)** **references training(t_id),** **foreign key (e_id)** **references evilness(e_id),** **primary key(t_id,e_id)** **);**
	training(t_id) **evilness(e_id)**	training	t_id	

ER COMPONENT	TYPE/ PRIMARY KEY(S)	RELATED ENTITIES & PRIMARY KEY		CORRESPONDING RELATION SCHEMA
	multi valued attribute	minion	m_id	**Create table languages_known(m_id number(4), language varchar(20), primary key(m_id,language), foreign key(m_id) references minion(m_id));**
	language **minion(m_id)**			

MINION DATABASE SCHEMA