

Converting (E)ER to Relational Mapping

Step 1: Mapping of regular(strong) entity types



all_users

VARCHAR(32)	VARCHAR(32)	VARCHAR(64)	VARCHAR(50)	VARCHAR(64)	VARCHAR(12)	INT()	DATE()	TINYINT(1)	TINYINT(1)
api_key	first_name	last_name	email	password	salt	num_reviews	last_verified	certified_critc	is_manager

wines

INT()	VARCHAR(50)	VARCHAR(64)	VARCHAR(255)	VARCHAR(255)	FLOAT()	TINYINT(1)	FLOAT()	VARCHAR(256)
wine_id	type	grape_type	image	description	price	quality	alcohol	designation

wineries

INT()	VARCHAR(64)	TINYINT(1)	TINYINT(1)	TINYINT(1)	TINYINT()
winery_id	winery_name	certified	operational	offers_tours	eco_friendly

Step 2: Mapping of weak entity type



all_users

VARCHAR(32)	VARCHAR(32)	VARCHAR(64)	VARCHAR(50)	VARCHAR(64)	VARCHAR(12)	INT()	DATE()	TINYINT(1)	TINYINT(1)
api_key	first_name	last_name	email	password	salt	num_reviews	last_verified	certified_critc	is_manager

reviews

VARCHAR(32)	INT()	INT()	FLOAT()
api_key	wine_id	review_id	rating

wines

INT()	VARCHAR(50)	VARCHAR(64)	VARCHAR(255)	VARCHAR(255)	FLOAT()	TINYINT(1)	FLOAT()	VARCHAR(256)
wine_id	type	grape_type	image	description	price	quality	alcohol	designation

wineries

INT()	VARCHAR(64)	TINYINT(1)	TINYINT(1)	TINYINT(1)	TINYINT()
winery_id	winery_name	certified	operational	offers_tours	eco_friendly

Step 3: Mapping binary (1:1) relationships

Approach take : Foreign key approach (approach 1)



all_users

VARCHAR(32)	VARCHAR(32)	VARCHAR(64)	VARCHAR(50)	VARCHAR(64)	VARCHAR(12)	INT()	DATE()	TINYINT(1)	TINYINT(1)
api_key	first_name	last_name	email	password	salt	num_reviews	last_verified	certified_critc	is_manager

reviews

VARCHAR(32)	INT()	INT()	FLOAT()
api_key	wine_id	review_id	rating

wines

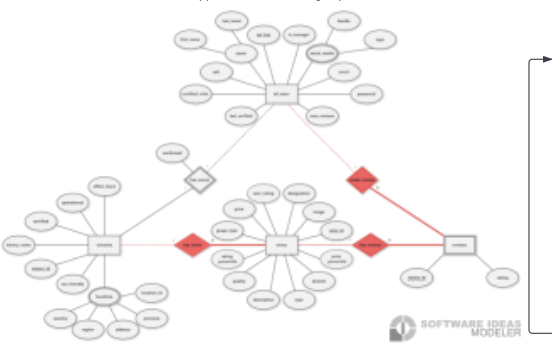
INT()	VARCHAR(50)	VARCHAR(64)	VARCHAR(255)	VARCHAR(255)	FLOAT()	TINYINT(1)	FLOAT()	VARCHAR(256)
wine_id	type	grape_type	image	description	price	quality	alcohol	designation

wineries

INT()	VARCHAR(32)	TINYINT(1)	VARCHAR(64)	TINYINT(1)	TINYINT(1)	TINYINT(1)	TINYINT()
winery_id	api_key	confirmed	winery_name	certified	operational	offers_tours	eco_friendly

Step 4: Mapping binary (1:N) relationships

Second approach take: creating separate relation for 1:N



all_users

VARCHAR(32)	VARCHAR(32)	VARCHAR(64)	VARCHAR(50)	VARCHAR(64)	VARCHAR(12)	INT()	DATE()	TINYINT(1)	TINYINT(1)
api_key	first_name	last_name	email	password	salt	num_reviews	last_verified	certified_critc	is_manager

reviews

VARCHAR(32)	INT()	INT()	FLOAT()	VARCHAR(255)
api_key	wine_id	review_id	rating	comment

wines

INT()	INT()	VARCHAR(50)	VARCHAR(64)	VARCHAR(255)	VARCHAR(255)	FLOAT()	TINYINT(1)	FLOAT()	VARCHAR(256)
wine_id	winery_id	type	grape_type	image	description	price	quality	alcohol	designation

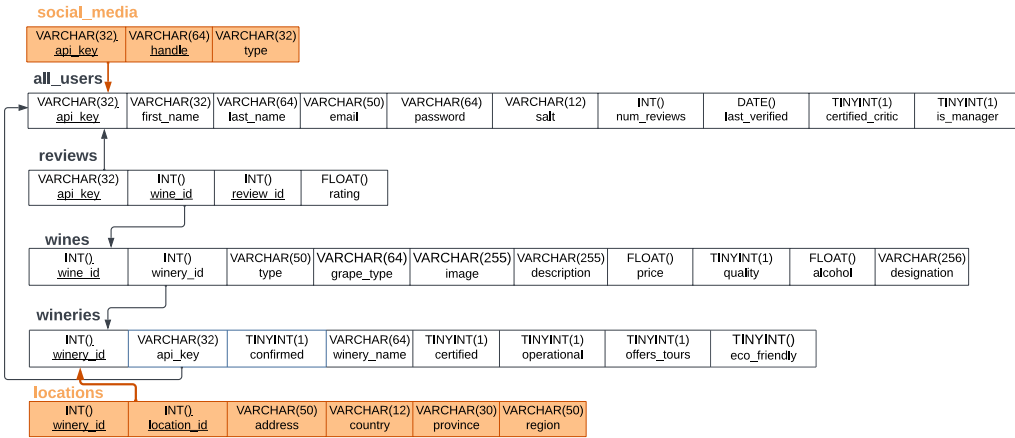
wineries

INT()	VARCHAR(32)	TINYINT(1)	VARCHAR(64)	TINYINT(1)	TINYINT(1)	TINYINT(1)	TINYINT()
winery_id	api_key	confirmed	winery_name	certified	operational	offers_tours	eco_friendly

Step 5: Mapping binary (M:N) relationships

N/A no M:N relationships need to be mapped

Step 6: Mapping multivalued attributes



Step 7: Mapping N-ary relationships

N/A No N-ary relationships need to be mapped

Step 8: Mapping Specialization and generalization

N/A No specialization/generalization needed to be mapped

Final Mapping:

