Database		listApp	CREATE DATABASE IF NOT EXISTS listApp; USE listApp; ALTER DATABASE listApp CHARACTER SET utf8 COLLATION utf8-general-ci
Table		user	CREATE TABLE IF NOT EXISTS users(id SMALLINT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT, username VARCHAR(255) NOT NULL, password VARCHAR(255) NOT NULL, email VARCHAR(320), home_collection_id INT UNSIGNED);
	PK	id	
		username	
		password //255 for the password_hash() improving by PHP	
		email	
		home_collection_id	
Table		collections	CREATE TABLE IF NOT EXISTS collections(id INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT, title VARCHAR(255) NOT NULL, addTime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP, author_id SMALLINT UNSIGNED NOT NULL,);
	PK	id	
		title	
		addtime	
	FK	author_id(user.id)	
Table		items	CREATE TABLE IF NOT EXISTS items(id INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT, title VARCHAR(255) NOT NULL, note TEXT, addTime TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP, type TINYINT UNSIGNED NOT NULL, author_id SMALLINT UNSIGNED NOT NULL, CONSTRAINT `fk_item_author` FOREIGN KEY (author_id) REFERENCES user (id) ON DELETE CASCADE);
	PK	id	
		title	
		note	
		addtime	
	FK	type (type_id)	
	FK	author_id(user.id)	
Table		checks	CREATE TABLE IF NOT EXISTS checks(item_id INT UNSIGNED NOT NULL PRIMARY KEY, //check 是保留字?? 在 phpmyadmin 新增成功 checked BOOLEAN NOT NULL DEFAULT false, CONSTRAINT `fk_check_item` FOREIGN KEY (item_id) REFERENCES items (id) ON DELETE CASCADE);
	PK/ FK	item_id (items.id)	
		checked	
Table		tasks	CREATE TABLE IF NOT EXISTS tasks(item_id INT UNSIGNED NOT NULL PRIMARY KEY, schedule DATETIME, due DATETIME, startTime DATETIME,
	PK/ FK	item_id (items.id)	
		schedule	

		due timer totalTime	endTime DATETIME, CONSTRAINT `fk_task_item` FOREIGN KEY (item_id) REFERENCES items (id) ON DELETE CASCADE);
Table		item_items	CREATE TABLE IF NOT EXISTS lists(id INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT, head_id INT UNSIGNED NOT NULL, type_id TINYINT UNSIGNED NOT NULL, CONSTRAINT `fk_list_head_item` FOREIGN KEY (head_id) REFERENCES items (id) ON DELETE CASCADE);
	PK FK FK	id head_id(items.id) type_id	

Table		item_items	CREATE TABLE IF NOT EXISTS item_items(parent_item INT UNSIGNED NOT NULL , child_item INT UNSIGNED NOT NULL, ordinal_num SMALLINT UNSIGNED, PRIMARY KEY(`parent_item`, `child_item`), CONSTRAINT `fk_itemri_items` FOREIGN KEY (parent_item) REFERENCES items (id) ON DELETE CASCADE, CONSTRAINT `fk_iritems_items` FOREIGN KEY (child_item) REFERENCES items (id) ON DELETE CASCADE);
	PK/ FK PK/ FK	parent_item child_item ordinal_num	
Table		collection_collections	CREATE TABLE IF NOT EXISTS collection_collections(parent_collection INT UNSIGNED NOT NULL , child_collection INT UNSIGNED NOT NULL, ordinal_num SMALLINT UNSIGNED, PRIMARY KEY(`parent_collection`, `child_collection`), CONSTRAINT `fk_collectionrc_collections` FOREIGN KEY (parent_collection) REFERENCES collections (id) ON DELETE CASCADE, CONSTRAINT `fk_crcollections_collections` FOREIGN KEY (child_collection) REFERENCES collections (id) ON DELETE CASCADE);
	PK/ FK PK/ FK	parent_collection child_collection	
Table		collection_items	CREATE TABLE IF NOT EXISTS collection_items(parent_collection INT UNSIGNED NOT NULL , child_item INT UNSIGNED NOT NULL, ordinal_num SMALLINT UNSIGNED, PRIMARY KEY(`parent_collection`, `child_item`), CONSTRAINT `fk_collectionri_collections` FOREIGN KEY (parent_collection) REFERENCES collections (id) ON DELETE CASCADE, CONSTRAINT `fk_critems_items` FOREIGN KEY (child_item) REFERENCES items (id) ON DELETE CASCADE
	PK/ FK PK/ FK	parent_item child_collection	

			);
Table		item_types	CREATE TABLE IF NOT EXISTS item_types(id TINYINT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT, `type_name` CHAR(50) NOT NULL);
	PK	id type_name	
		list_types	CREATE TABLE IF NOT EXISTS list_types(id TINYINT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT, type_name CHAR(50) NOT NULL, head_type_id TINYINT UNSIGNED NOT NULL, new_items_type_id TINYINT UNSIGNED NOT NULL);
	PK	id type_name	
	FK	head_type_id	
	FK	new_items_type_id	