Schéma : dellstore

dellstore, relationnel, pattes de corbeau, crowfoot

2024-09-20

⚠ Présentation du schéma `dellstore`, pour l'entrainement à PostGres.

- **L3 MIASHS/Ingémath**
- **Université Paris Cité**
- Année 2024-2025
- Course Homepage
- Moodle

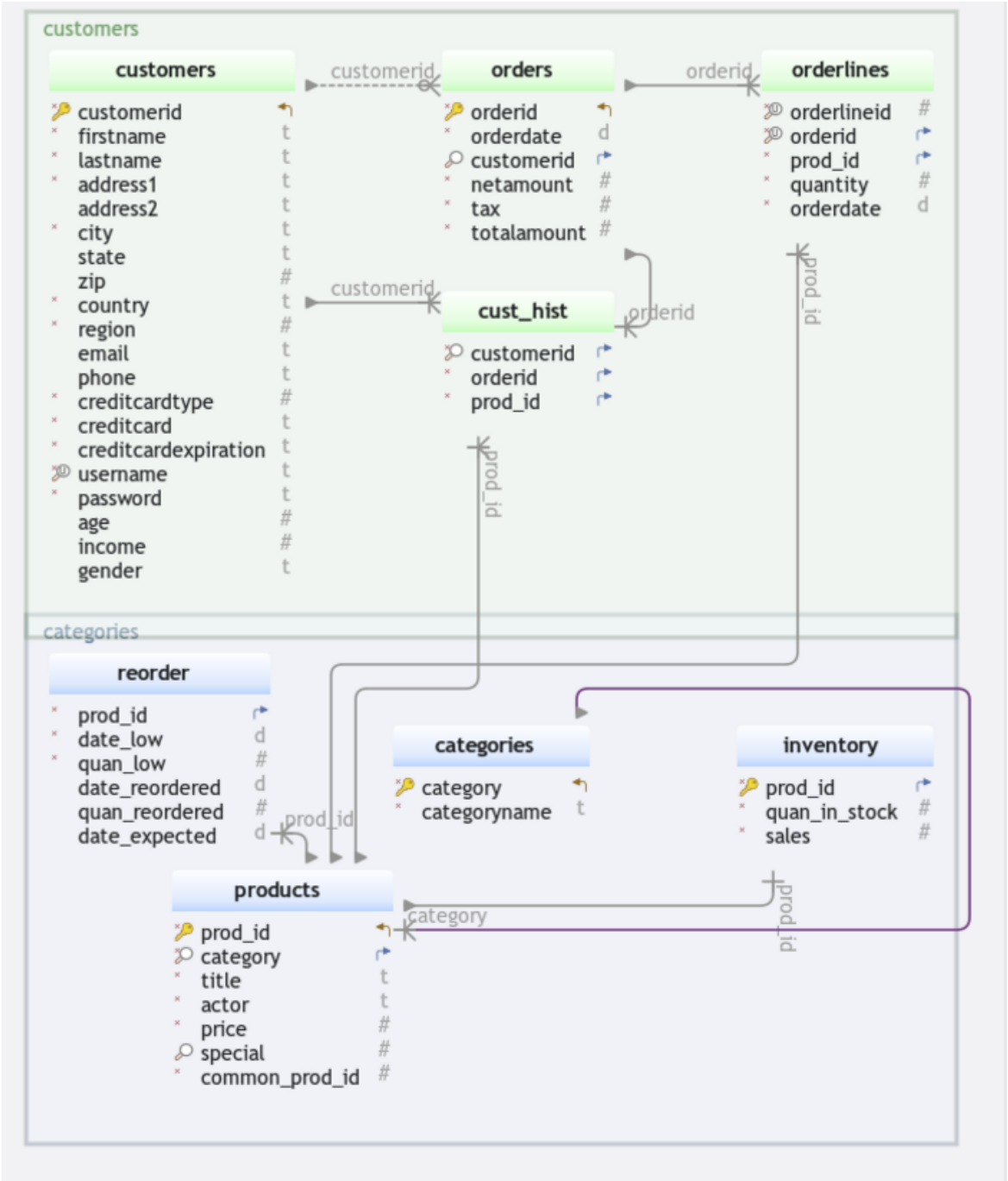

Fig. 1 : Schema dellstore créé par DbSchema

Contexte

La base de données du magasin de DVD est gérée par six procédures stockées. Les deux premières sont utilisées pendant la phase de connexion.

Si le client revient, la procédure Login est utilisée pour récupérer les informations du client, en particulier l'identifiant du client (CUSTOMERID). Si le client est un nouveau client, New_customer est utilisée pour créer une nouvelle ligne dans la table Customers avec les données de l'utilisateur. Après la phase de connexion, le client peut rechercher un DVD par catégorie, acteur ou titre, acteur ou titre. Ces critères sont mis en œuvre par Browse_by_category, Browse_by_actor et Browse_by_title, respectivement. Enfin, une fois que l'utilisateur a fait ses choix, la procédure stockée Purchase est appelée pour terminer la transaction.

Les procédures stockées de l'application DS2 présentent des caractéristiques qui permettent de mieux modéliser les magasins en ligne d'aujourd'hui.

Lors de la connexion, par exemple, la commande précédente de l'utilisateur (jusqu'à dix titres) est signalée, ainsi que les titres recommandés par d'autres clients qui aiment ces titres.

La fonction Parcourir_par_catégorie renvoie les titres de la catégorie spécifiée qui sont actuellement en vente.

Enfin, la procédure stockée Achat vérifie désormais le champ QUAN_IN_STOCK de la table Inventaire pour voir si un titre est disponible. Cette opération est réalisée à l'aide d'une transaction de base de données,

de sorte que si la quantité est insuffisante pour honorer la commande, les données QUAN_IN_STOCK ne sont pas mises à jour et un nouvel enregistrement n'est pas écrit dans la base de données. ni un nouvel enregistrement dans la table ORDERS.

Le schéma de la base de données « dellstore » dans PostgreSQL, utilisée à des fins de formation.

1. **Tables** :
   - **dellstore.categories** : Contient les détails des catégories avec des colonnes comme `category` (clé primaire) et `categoryname`. (`Action`, `Animation`, `Children`, …, en tout 16 catégories)
   - **dellstore.cust_hist** : Trace l'*historique* des clients avec des références aux clients , aux commandes (*orders*)et aux produits (*products*), en utilisant des clés étrangères (`customerid`, `(*customers*)orderid`, et `prod_id`).
   - **dellstore.customers** : Stocke les informations sur les clients (*customers*) avec des champs tels que `customerid`, `firstname`, `lastname`, `address`, `city`, et `creditcard` details.
   - **dellstore.inventory** : Gère l'inventaire des produits en suivant `prod_id` (clé primaire).
2. **Clés étrangères** :
   - **dellstore.cust_hist** a des références de clés étrangères à d'autres tables comme `customers`, `orders`, et `products`.

Ce schéma décrit les tables clés utilisées pour une application de type commerce électronique, gérant les catégories, les informations sur les clients et les transactions. Vous souhaitez obtenir une description plus détaillée de certaines tables ou de leurs relations ?

Les sources du schéma sont décrites sur le site de Dell

Table dellstore.categories

Idx	Name	Data Type
*	category	integer DEFAULT nextval('dellstore.categories_category_seq' ::regclass)
*	categoryname	varchar(50)

Table dellstore.cust_hist

Idx	Name	Data Type
*	customerid	integer
*	orderid	integer
*	prod_id	integer

Foreign Keys

Type	Name	On
	fk_cust_hist_customerid	(customerid) ref dellstore.customers (customerid)
	fk_cust_hist_orders	(orderid) ref dellstore.orders (orderid)

Type	Name	On
	fk_cust_hist_products	(prod_id) ref dellstore.products (prod_id)

👉 C'est une table *raccourci* (redondante) dont le contenu peut être reconstruit à l'aide de `customers`, `orders` et `orderlines` ?

Table dellstore.customers

Idx	Name	Data Type
*	customerid	integer DEFAULT nextval('dellstore.customers_customerid_seq'::regclass)
*	firstname	varchar(50)
*	lastname	varchar(50)
*	address1	varchar(50)
	address2	varchar(50)
*	city	varchar(50)
	state	varchar(50)
	zip	integer
*	country	varchar(50)
*	region	smallint
	email	varchar(50)
	phone	varchar(50)
*	creditcardtype	integer
*	creditcard	varchar(50)
*	creditcardexpiration	varchar(50)
*	username	varchar(50)
*	password	varchar(50)
	age	smallint
	income	integer
	gender	varchar(1)

Table dellstore.inventory

Idx	Name	Data Type
*	prod_id	integer
*	quan_in_stock	integer
*	sales	integer

Foreign Keys

Type	Name	On
	fk_inventory_products	(prod_id) ref dellstore.products (prod_id)

Cette table nous indique l'état du stock pour chaque produit au catalogue, et combien d'exemplaires du produit ont étés vendus.

Table dellstore.orderlines

Idx	Name	Data Type
*	orderlineid	integer
*	orderid	integer
*	prod_id	integer
*	quantity	smallint
*	orderdate	date

Foreign Keys

Type	Name	On
	fk_orderid	(orderid) ref dellstore.orders (orderid)
	fk_orderlines_products	(prod_id) ref dellstore.products (prod_id)

Chaque ligne de `orderlines` nous renseigne sur l'achat d'un produit désigné par `prod_id` en une certaine quantité , à une certaine date.

Table dellstore.orders

Idx	Name	Data Type
*	orderid	integer DEFAULT nextval('dellstore.orders_orderid_seq'::regclass)
*	orderdate	date
	customerid	integer
*	netamount	numeric(12,2)
*	tax	numeric(12,2)
*	totalamount	numeric(12,2)

Foreign Keys

Type	Name	On
	fk_customerid	(customerid) ref dellstore.customers (customerid)

Table dellstore.products

Idx	Name	Data Type
*	prod_id	integer DEFAULT nextval('dellstore.products_prod_id_seq'::regclass)
*	category	integer
*	title	varchar(50)
*	actor	varchar(50)
*	price	numeric(12,2)
	special	smallint
*	common_prod_id	integer

Un petit apperçu de la table aide à comprendre :

```
+---------+----------+--------------------+------------------+-------+---------+----------------+
| prod_id | category | title              | actor            | price | special | common_prod_id |
|---------+----------+--------------------+------------------+-------+---------+----------------|
| 1       | 14       | ACADEMY ACADEMY    | PENELOPE GUINESS | 25.99 | 0       | 1976           |
| 2       | 6        | ACADEMY ACE        | EWAN RICKMAN     | 20.99 | 0       | 6289           |
| 3       | 6        | ACADEMY ADAPTATION | VIVIEN KAHN      | 28.99 | 0       | 7173           |
| 4       | 3        | ACADEMY AFFAIR     | ALAN MARX        | 14.99 | 0       | 8042           |
| 5       | 3        | ACADEMY AFRICAN    | CARRIE HANNAH    | 11.99 | 1       | 2183           |
| 6       | 9        | ACADEMY AGENT      | LISA SPACEK      | 15.99 | 0       | 5243           |
| 7       | 8        | ACADEMY AIRPLANE   | FRANCES WINFREY  | 25.99 | 0       | 7700           |
| 8       | 7        | ACADEMY AIRPORT    | FARRAH TOMEI     | 16.99 | 0       | 9191           |
| 9       | 2        | ACADEMY ALABAMA    | JULIETTE WEAVER  | 10.99 | 0       | 6633           |
| 10      | 15       | ACADEMY ALADDIN    | ANNETTE FREEMAN  | 9.99  | 0       | 631            |
+---------+----------+--------------------+------------------+-------+---------+----------------+
```

Une ligne représente un DVD fictif. Un DVD contient un film désigné par un titre. Dans ce film apparaît une actrice ou un acteur. Le DVD est vendu au prix `price` (USD ?). Un film/produi relève d'une seule catégorie. - `special` vaut 1 pour 104 lignes, 0 pour les autes. - `common_prod_id` ?

Foreign Keys

Type	Name	On
Vir	fk_products_category	(category) ref dellstore.categories (category)

Table dellstore.reorder

Idx	Name	Data Type
*	prod_id	integer
*	date_low	date
*	quan_low	integer
	date_reordered	date
	quan_reordered	integer
	date_expected	date

Foreign Keys

Type	Name	On
	fk_reorder_products	(prod_id) ref dellstore.products (prod_id)

> On peut se demander quel est l'usage de la table `reorder`.

La table `reorder` gère (peut-être) les réapprovisionements de produits (lorsque l'inventaire `inventory` signale la quantité en stock d'un produit est dangereusement basse, on effectue une nouvelle commande).

- Colonnes :
  - `prod_id` : un nombre entier identifiant le produit (clé étrangère liée à la table `products`).
  - `date_low` : la date à laquelle le niveau de stock a été détecté comme bas.
  - `quan_low` : La quantité qui a déclenché le processus de réapprovisionnement.
  - `date_reordered` : La date à laquelle le produit a été commandé à nouveau.
  - `quan_reordered` : La quantité du produit commandé à nouveau.
  - `date_expected` : la date à laquelle les produits commandés à nouveau sont censés arriver.
- ** Foreign Key ** :

– `fk_reorder_products` : lie `prod_id` à la table `products`.

👉 Dans l'instance courante du schéma, cette table est vide.