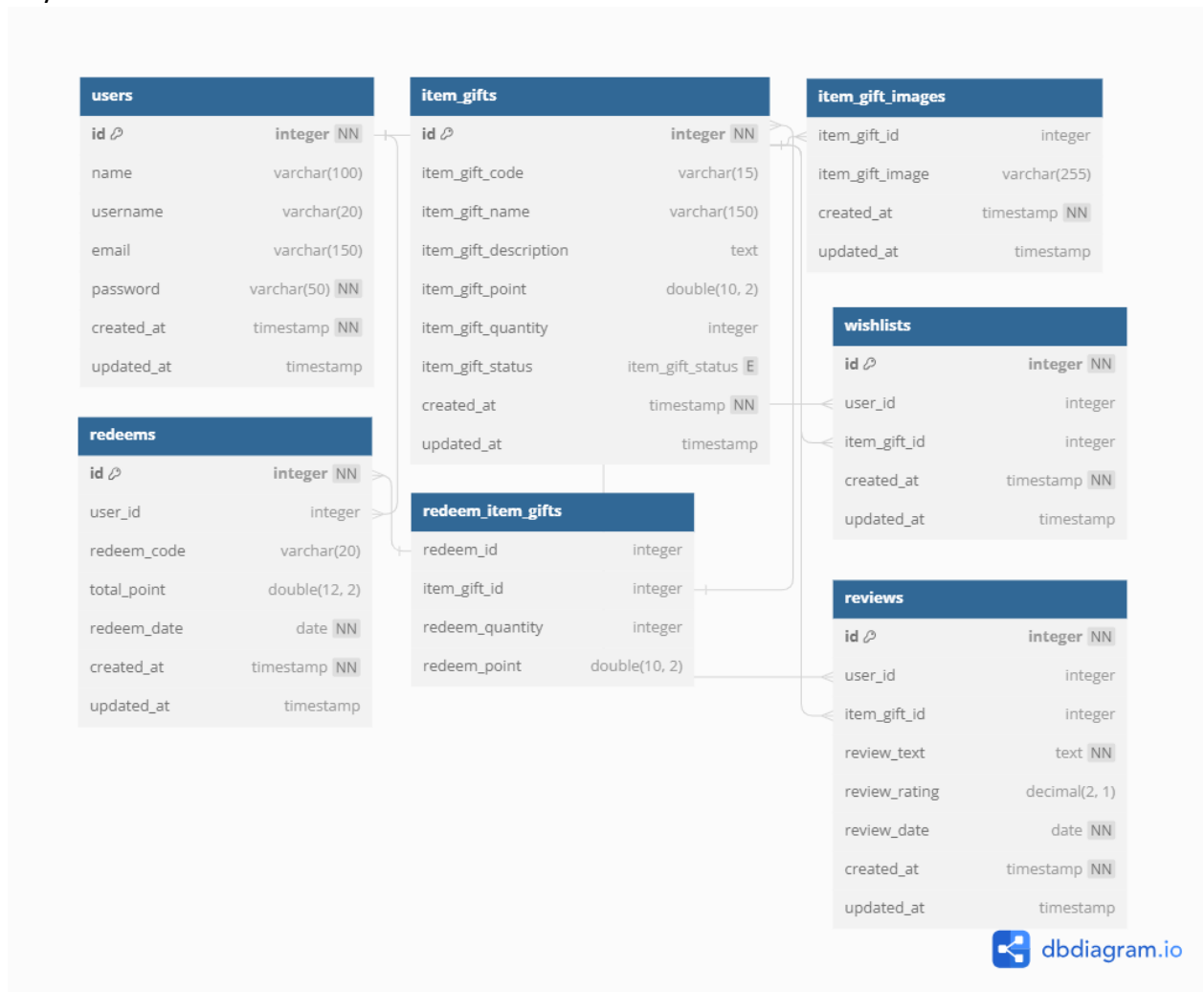


Documentation

1. Physical ERD



Tabel yang ada dalam skema adalah sebagai berikut:

- **Tabel users:**

1. id: integer, primary key, not null. Kolom ini merupakan *primary key* untuk setiap user.
2. name: varchar(100). Kolom ini menyimpan nama user.
3. username: varchar(20), unique. Kolom ini menyimpan username user secara unik.
4. email: varchar(150), unique. Kolom ini menyimpan alamat email user secara unik.
5. password: varchar(50), not null. Kolom ini menyimpan kata sandi user.
6. created_at: timestamp, not null, default: now(). Kolom ini menyimpan tanggal dan waktu pembuatan user.

7. `updated_at`: timestamp. Kolom ini menyimpan tanggal dan waktu terakhir user diperbarui.

- **Tabel `item_gifts`:**

1. `id`: integer, primary key, not null. Kolom ini merupakan *primary key* untuk setiap item gift.
2. `item_gift_code`: varchar(15), unique. Kolom ini menyimpan kode unik untuk setiap item gift.
3. `item_gift_name`: varchar(150). Kolom ini menyimpan nama item gift.
4. `item_gift_description`: text. Kolom ini menyimpan deskripsi item gift.
5. `item_gift_point`: double(10, 2). Kolom ini menyimpan jumlah poin yang diperlukan untuk mendapatkan item gift.
6. `item_gift_quantity`: integer. Kolom ini menyimpan jumlah kuantitas item gift yang tersedia.
7. `item_gift_status`: enum `item_gift_status`. Kolom ini menyimpan status item gift, yang dapat berupa "Available" (A) atau "Out Of Stock" (O).
8. `created_at`: timestamp, not null, default: `now()`. Kolom ini menyimpan tanggal dan waktu pembuatan item gift.
9. `updated_at`: timestamp. Kolom ini menyimpan tanggal dan waktu terakhir item gift diperbarui.

Indeks:

1. `item_gift_code`: Indeks pada kolom `item_gift_code` untuk pencarian dan pengurutan berdasarkan kode item gift.
2. `item_gift_name`: Indeks pada kolom `item_gift_name` untuk pencarian dan pengurutan berdasarkan nama item gift.

- **Tabel `item_gift_images`:**

1. `item_gift_id`: integer. Kolom ini adalah *foreign key* ke tabel `item_gifts` untuk menunjukkan item gift yang terkait dengan gambar.
2. `item_gift_image`: varchar(255). Kolom ini menyimpan nama gambar untuk item gift.
3. `created_at`: timestamp, not null, default: `now()`. Kolom ini menyimpan tanggal dan waktu pembuatan gambar item gift.
4. `updated_at`: timestamp. Kolom ini menyimpan tanggal dan waktu terakhir gambar item gift diperbarui.

Indeks:

1. `item_gift_id`: Indeks pada kolom `item_gift_id` untuk pencarian dan pengurutan berdasarkan item gift.

- **Tabel `redeems`:**

1. `id`: integer, primary key, not null. Kolom ini merupakan *primary key* untuk setiap redeem.

2. `user_id`: integer. Kolom ini adalah *foreign key* ke tabel `users` untuk menunjukkan user yang melakukan redeem.
3. `redeem_code`: `varchar(20)`, unique. Kolom ini menyimpan kode unik untuk setiap redeem.
4. `total_point`: `double(12, 2)`. Kolom ini menyimpan jumlah total poin yang di redeem.
5. `redeem_date`: date, not null. Kolom ini menyimpan tanggal redeem.
6. `created_at`: timestamp, not null, default: `now()`. Kolom ini menyimpan tanggal dan waktu pembuatan redeem.
7. `updated_at`: timestamp. Kolom ini menyimpan tanggal dan waktu terakhir redeem diperbarui.

Indeks:

1. `user_id`: Indeks pada kolom `user_id` untuk pencarian dan pengurutan berdasarkan user.
2. `redeem_code`: Indeks pada kolom `redeem_code` untuk pencarian dan pengurutan berdasarkan kode redeem.
3. `redeem_date`: Indeks pada kolom `redeem_date` untuk pencarian dan pengurutan berdasarkan tanggal redeem.

- **Tabel `redeem_item_gifts`:**

1. `redeem_id`: integer. Kolom ini adalah *foreign key* ke tabel `redeems` untuk menunjukkan redeem yang terkait dengan item gift.
2. `item_gift_id`: integer. Kolom ini adalah *foreign key* ke tabel `item_gifts` untuk menunjukkan item gift yang di redeem.
3. `redeem_quantity`: integer. Kolom ini menyimpan jumlah kuantitas item gift yang di redeem dalam satu proses redeem.
4. `redeem_point`: `double(10, 2)`. Kolom ini menyimpan jumlah poin yang di redeem untuk item gift.

Indeks:

1. `redeem_id`: Indeks pada kolom `redeem_id` untuk pencarian dan pengurutan berdasarkan redeem.
2. `item_gift_id`: Indeks pada kolom `item_gift_id` untuk pencarian dan pengurutan berdasarkan item gift.

- **Tabel `reviews`:**

1. `id`: integer, primary key, not null. Kolom ini merupakan *primary key* untuk setiap ulasan.
2. `user_id`: integer. Kolom ini adalah *foreign key* ke tabel `users` untuk menunjukkan pengguna yang melakukan ulasan.
3. `item_gift_id`: integer. Kolom ini adalah *foreign key* ke tabel `item_gifts` untuk menunjukkan item gift yang diulas.
4. `review_text`: text, not null. Kolom ini menyimpan teks ulasan.

5. `review_rating`: decimal(2, 1). Kolom ini menyimpan nilai peringkat ulasan.
6. `review_date`: date, not null. Kolom ini menyimpan tanggal ulasan.
7. `created_at`: timestamp, not null, default: `now()`. Kolom ini menyimpan tanggal dan waktu pembuatan ulasan.
8. `updated_at`: timestamp. Kolom ini menyimpan tanggal dan waktu terakhir ulasan diperbarui.

Indeks:

1. `user_id`: Indeks pada kolom `user_id` untuk pencarian dan pengurutan berdasarkan user.
2. `item_gift_id`: Indeks pada kolom `item_gift_id` untuk pencarian dan pengurutan berdasarkan item gift.
3. `review_date`: Indeks pada kolom `review_date` untuk pencarian dan pengurutan berdasarkan tanggal ulasan.

- **Tabel wishlists:**

1. `id`: integer, primary key, not null. Kolom ini merupakan *primary key* untuk setiap wishlist.
2. `user_id`: integer. Kolom ini adalah *foreign key* ke tabel `users` untuk menunjukkan user yang memiliki wishlist.
3. `item_gift_id`: integer. Kolom ini adalah *foreign key* ke tabel `item_gifts` untuk menunjukkan item hadiah yang ada di wishlist.
4. `created_at`: timestamp, not null, default: `now()`. Kolom ini menyimpan tanggal dan waktu pembuatan wishlist.
5. `updated_at`: timestamp. Kolom ini menyimpan tanggal dan waktu terakhir wishlist diperbarui.

Indeks:

1. `user_id`: Indeks pada kolom `user_id` untuk pencarian dan pengurutan berdasarkan user.
2. `item_gift_id`: Indeks pada kolom `item_gift_id` untuk pencarian dan pengurutan berdasarkan item gift.

Relasi antara tabel-tabel adalah sebagai berikut:

- Tabel `redeems.id` > `redeem_item_gifts.redeem_id`: Relasi one-to-many antara tabel `redeems` dan `redeem_item_gifts`. Setiap `redeem` dapat memiliki beberapa item gift pada satu proses `redeem`.
- Tabel `item_gifts.id` > `redeem_item_gifts.item_gift_id`: Relasi one-to-many antara tabel `item_gifts` dan `redeem_item_gifts`. Setiap item gift dapat di `redeem` dalam beberapa proses `redeem`.
- Tabel `item_gift_images.item_gift_id` > `item_gifts.id`: Relasi one-to-many antara tabel `item_gifts` dan `item_gift_images`. Setiap item gift dapat memiliki lebih satu gambar yang terkait.

- Tabel `users.id` < `redeems.user_id`: Relasi one-to-many antara tabel `users` dan `redeems`. Setiap user dapat melakukan beberapa redeem.
- Tabel `users.id` < `reviews.user_id`: Relasi one-to-many antara tabel `users` dan `reviews`. Setiap user dapat melakukan beberapa ulasan.
- Tabel `item_gifts.id` < `reviews.item_gift_id`: Relasi one-to-many antara tabel `item_gifts` dan `reviews`. Setiap item gift dapat memiliki beberapa ulasan.
- Tabel `users.id` < `wishlists.user_id`: Relasi one-to-many antara tabel `users` dan `wishlists`. Setiap user dapat memiliki beberapa wishlist.
- Tabel `item_gifts.id` < `wishlists.item_gift_id`: Relasi one-to-many antara tabel `item_gifts` dan `wishlists`. Setiap item gift dapat ada dalam beberapa wishlist.

2. Alasan saya menggunakan Laravel

- Laravel adalah salah satu framework PHP yang populer dan powerful. MySQL menyediakan banyak fitur dan alat bantu untuk membangun aplikasi web secara efisien.
- Laravel menggunakan konsep MVC (Model-View-Controller) yang membantu dalam pemisahan logika bisnis, tampilan, dan logika routing sehingga membuat kode lebih terorganisir dan mudah di-maintain.
- Laravel memiliki sintaks yang ekspresif dan mudah dipahami, serta menyediakan banyak library dan paket yang siap digunakan, seperti autentikasi, validasi, caching, dan lainnya. Hal ini mempercepat proses pengembangan aplikasi.
- Laravel juga memiliki fitur ORM (Object-Relational Mapping) yang kuat bernama Eloquent, yang menyederhanakan interaksi dengan database dan menghilangkan perluasan SQL secara langsung.

3. Saya menggunakan Arsitektur Repository Pattern

Arsitektur Repository Pattern membantu dalam memisahkan logika bisnis dari logika akses data, memungkinkan kode yang lebih terorganisir, modular, dan mudah diuji. Hal ini data mempermudah pemeliharaan, pengembangan, dan skalabilitas aplikasi yang lebih baik.

4. Alasan saya memilih MySQL

- MySQL adalah salah satu sistem manajemen basis data relasional (RDBMS) yang paling populer dan banyak digunakan di dunia.
- MySQL memiliki kinerja yang baik dan dapat menangani beban kerja yang tinggi dengan cepat.
- MySQL kompatibel dengan sebagian besar platform dan memiliki dukungan yang luas dalam komunitas pengembang.
- MySQL menyediakan fitur-fitur seperti dukungan transaksi ACID (Atomicity, Consistency, Isolation, Durability), indeks yang kuat, keamanan yang baik, dan kemampuan replikasi untuk skalabilitas dan redundansi.

5. Berikut adalah Optimasi database yang saya lakukan:

- *Indexing*: Dengan melakukan indexing pada kolom-kolom yang sering digunakan dalam operasi pencarian atau pemfilteran data, query dapat dieksekusi dengan lebih efisien. Index mempercepat proses pencarian dan pengurutan data.
- *Normalisasi*: Dalam perancangan basis data, normalisasi membantu menghindari redundansi data dan memperbaiki struktur tabel. Ini mengurangi ukuran data dan meningkatkan konsistensi serta integritas data.
- *Query Optimization*: Melakukan optimasi query dengan menggunakan teknik seperti penggunaan indeks yang tepat, menulis query yang efisien, menghindari penggunaan * dalam SELECT, dan menggunakan join dengan bijak.

6. Alasan saya memilih AWS

- AWS (Amazon Web Services) adalah salah satu penyedia layanan cloud computing terbesar di dunia, dengan infrastruktur yang kuat dan stabil.
- AWS menawarkan berbagai layanan yang sangat fleksibel, seperti cloud server (EC2), penyimpanan (S3), basis data (RDS), dan lain-lain. Hal ini memungkinkan skalabilitas dan elastisitas yang tinggi untuk aplikasi.
- AWS memiliki global infrastruktur yang luas, dengan pusat data yang tersebar di berbagai wilayah di seluruh dunia. Ini memungkinkan aplikasi dapat di-deploy di dekat pengguna untuk mengurangi latensi dan meningkatkan kinerja.
- AWS memiliki fitur keamanan yang kuat dan menyediakan alat pengelolaan akses dan enkripsi data yang dapat membantu melindungi aplikasi dan data.
- AWS juga menawarkan berbagai alat dan layanan tambahan seperti pemantauan (CloudWatch), manajemen log (CloudTrail), manajemen aplikasi (Elastic Beanstalk), dan lainnya yang dapat memperkaya pengalaman pengembangan dan operasional aplikasi.