## 예제: Enum 타입 저장 및 불러오기

우리는 사용자 레벨을 나타내는 `UserLevel`이라는 Enum 클래스를 사용한다고 가정해보겠습니다.

### UserLevel Enum 정의

kotlin 코드 복사

```kotlin
enum class UserLevel {
    NORMAL,
    ADMIN,
    SUPERUSER
}
```

### User 엔티티 클래스 정의

이제 `User` 엔티티 클래스에 `UserLevel` 타입의 속성을 추가합니다.

kotlin 코드 복사

```kotlin
@Entity(tableName = "user")
data class User(
    @PrimaryKey val id: Int,
    val name: String,
    val level: UserLevel
)
```

### 타입 컨버터 정의

Room은 기본적으로 Enum 타입을 데이터베이스에 저장할 수 없으므로, Enum을 문자열이나 정수로 변환하는 타입 컨버터를 정의해야 합니다.

kotlin 코드 복사

```kotlin
class UserLevelConverter {
    @TypeConverter
    fun fromUserLevel(value: UserLevel?): String? {
        return value?.name
    }

    @TypeConverter
    fun toUserLevel(value: String?): UserLevel? {
        return value?.let { UserLevel.valueOf(it) }
    }
}
```

설명:

- `fromUserLevel(value: UserLevel?)`: `UserLevel` Enum 값을 문자열로 변환합니다.
- `toUserLevel(value: String?)`: 문자열 값을 `UserLevel` Enum 값으로 변환합니다.

데이터베이스 클래스에 타입 컨버터 등록

kotlin 코드 복사

```kotlin
@Database(
    entities = [User::class],
    version = 1
)
@TypeConverters(UserLevelConverter::class)
abstract class ApplicationDatabase: RoomDatabase() {
    abstract fun userDao(): UserDao
}
```

UserDao 정의

kotlin 코드 복사

```kotlin
@Dao
interface UserDao {
    @Insert
    fun insertUser(user: User)

    @Query("SELECT * FROM user WHERE id = :id")
    fun getUserById(id: Int): User?
}
```

```
// Enum 클래스 정의
enum class UserLevel {
    NORMAL,
    ADMIN,
    SUPERUSER
}

// 엔티티 클래스 정의
```

```
@Entity(tableName = "user")
data class User(
    @PrimaryKey val id: Int,
    val name: String,
    val level: UserLevel
)

// 타입 컨버터 정의
class UserLevelConverter {
    @TypeConverter
    fun fromUserLevel(value: UserLevel?): String? {
        return value?.name
    }

    @TypeConverter
    fun toUserLevel(value: String?): UserLevel? {
        return value?.let { UserLevel.valueOf(it) }
    }
}

// 데이터베이스 클래스 정의 및 타입 컨버터 등록
@Database(
    entities = [User::class],
    version = 1
```

```
)
@TypeConverters(UserLevelConverter::class)
abstract class ApplicationDatabase: RoomDatabase() {
    abstract fun userDao(): UserDao
}

// DAO 정의
@Dao
interface UserDao {
    @Insert
    fun insertUser(user: User)

    @Query("SELECT * FROM user WHERE id = :id")
    fun getUserById(id: Int): User?
}
```