

DB 연동 결과

```
[notice] A new release of pip is available: 25.1.1 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
(env) PS C:\Users\yyson\Downloads\bedrock-lecture-code\bedrock-lecture-code\chapter4> python dbtest.py
PostgreSQL 버전: PostgreSQL 17.6 on aarch64-unknown-linux-gnu, compiled by gcc (GCC) 12.4.0, 64-bit
pgvector 버전: 0.8.0
```

DB 연결 성공!

```
(env) PS C:\Users\yyson\Downloads\bedrock-lecture-code\bedrock-lecture-code\chapter4> █
```

- tabase.pyC:\Users\yyson\Downloads\bedrock-lecture-code\bedrock-lecture-code\chapter4>
 1. 'database-1' 데이터베이스 생성 중...
데이터베이스가 생성되었습니다.
 2. 'database-1'에 pgvector 확장 설치 중...
pgvector 확장이 설치되었습니다.

모든 작업이 완료되었습니다!
이제 database='database-1'로 연결할 수 있습니다.

• (env) PS C:\Users\yyson\Downloads\bedrock-lecture-code\bedrock-lecture-code\chapter4> python check_db.py

사용 가능한 데이터베이스 목록을 확인합니다...

사용 가능한 데이터베이스:

-
- bedrock
 - database-1
 - postgres
 - rdsadmin

연결 성공!

● (env) PS C:\Users\yyson\Downloads\bedrock-lecture-code\bedrock-lecture-code\chapter4> python create_table.py
테이블을 생성합니다...

1. 'bedrock_integration' 스키마 생성 중...
스키마가 생성되었습니다.
2. 'bedrock_integration.bedrock_kb' 테이블 생성 중...
테이블이 생성되었습니다.
3. 벡터 검색을 위한 인덱스 생성 중...
인덱스가 생성되었습니다.
4. 생성된 테이블 구조 확인:
 - id: uuid
 - embedding: USER-DEFINED
 - chunks: text
 - metadata: json
 - created_at: timestamp without time zone

모든 작업이 완료되었습니다!

이제 bedrock_integration.bedrock_kb 테이블을 사용할 수 있습니다.

```
● (env) PS C:\Users\yyson\Downloads\bedrock-lecture-code\bedrock-lecture-code\chapter4> python insert_sample_data.py
샘플 데이터 삽입 중...
[1] 삽입 완료: Amazon Bedrock은 AI21 Labs, Anthropic, Cohere, Meta...
[2] 삽입 완료: Amazon Titan Embeddings는 텍스트를 수치 표현으로 변환하여 검색, 개인화...
[3] 삽입 완료: RAG(Retrieval-Augmented Generation)는 외부 지식 베이스를 활용...

모든 샘플 데이터 삽입 완료!

==== 삽입된 데이터 확인 ===
총 3개의 레코드
```

```
(env) PS C:\Users\yyson\Downloads\bedrock-lecture-code\bedrock-lecture-code\chapter4> python ./insert_sample_data_langchain.py
```

```
Bedrock Embeddings 모델 로딩 중...
```

```
샘플 데이터 5개를 벡터 DB에 삽입 중...
```

```
(Bedrock API를 통해 임베딩 생성 중...)
```

```
모든 샘플 데이터 삽입 완료!
```

- Collection: aws_vector
- 삽입된 문서: 5개
- 임베딩 차원: 1024 (Titan Embed Text V2)

```
==== 유사도 검색 테스트 ===
```

```
질문: Bedrock이란 무엇인가요?
```

```
가장 유사한 2개 문서:
```

```
[1] Amazon Bedrock은 AI21 Labs, Anthropic, Cohere, Meta, Stability AI, Amazon의 기초 모델(...)  
메타데이터: {'source': 'aws_docs', 'category': 'bedrock_intro', 'created_at': '2024-01-01'}
```

```
[2] Claude 3.5 Sonnet은 복잡한 추론과 코딩 작업에 뛰어난 성능을 보이는 Anthropic의 최신 모델입니다..
```

```
..  
메타데이터: {'source': 'anthropic_docs', 'category': 'claude_model', 'created_at': '2024-01-05'}
```

```
● (env) PS C:\Users\yyson\Downloads\bedrock-lecture-code\bedrock-lecture-code\chapter4> python .\db_select.py
```

데이터 조회를 시작합니다...

연결된 데이터베이스: bedrock @ database-1.cbuaymeeggmf.ap-northeast-2.rds.amazonaws.com

==== 1. 스키마 확인 ===

```
- {'schema_name': 'bedrock_integration'}
```

==== 2. 테이블 확인 (bedrock_integration) ===

```
- {'table_name': 'bedrock_kb'}
```

==== 3. 테이블 구조 확인 (bedrock_kb) ===

```
- {'column_name': 'id', 'data_type': 'uuid'}
- {'column_name': 'embedding',
- {'column_name': 'chunks', 'd
- {'column_name': 'metadata',
- {'column_name': 'created_at'
```

==== 4. 현재 데이터 개수 (bedrock_kb) ===

```
- {'count': 3}
```

==== 5. 샘플 데이터 조회 (최대 5개) ===

```
- {'id': 'a9c15adf-9e7b-44e0-a9ab-396816f308e9', 'chunks': 'Amazon Bedrock은 AI21 Labs, Anthropic, Cohere, Meta, Stability AI, Amazon의 기초 모델(FM)을 단일 API로 제공하는 완전관리형 서비스입니다.', 'metadata': {'source': 'aws_docs', 'category': 'bedrock_intro', 'created_at': '2024-01-01'}}
- {'id': '05d2ee77-131e-4f73-b0b3-01522959164f', 'chunks': 'Amazon Titan Embeddings는 텍스트를 수치 표현으로 변환하여 검색, 개인화 및 클러스터링과 같은 사용 사례에 유용합니다.', 'metadata': {'source': 'aws_docs', 'category': 'titan_embedding', 'created_at': '2024-01-02'}}
- {'id': '98b16977-d17c-48f1-b41c-219aab7a06bc', 'chunks': 'RAG(Retrieval-Augmented Generation)은 일부 지식 베이스를 활용하여 LLM의 응답 품질을 향상시키는 기술입니다.', 'metadata': {'source': 'technical_docs', 'category': 'rag_concept', 'created_at': '2024-01-03'}}
```

==== 6. pgvector 확장 확인 ===

```
- {'extname': 'vector', 'extversion': '0.8.0'}
```

==== 7. LangChain 컬렉션 목록 (langchain_pg_collection) ===

```
- {'name': 'aws_vector', 'cmetadata': None, 'uuid': 'c596f8cc-2881-48a6-b807-7e1a9f7c88b2'}
```

==== 8.a LangChain 임베딩 - 컬럼 확인 (langchain_pg_embedding) ===

```
- {'column_name': 'collection_id'}
- {'column_name': 'embedding'}
- {'column_name': 'cmetadata'}
- {'column_name': 'uuid'}
- {'column_name': 'document'}
```