what could be reason of error

chromadb.errors.InternalError: Error updating collection: Database error: error returned from database: (code: 1032) attempt to write a readonly database

```
            # Ingest pdf to vectordb
            if file_type == "pdf":
                ids = ingest_pdf_vectordb(
                    file_path=file_path
                )

# Create embedding model
EMBEDDING = get_embedding(
    llm_provider=cfg.LLM_PROVIDER,
    embedding_model=cfg.EMBEDDING_MODEL,
    api_key=cfg.OPENAI_API_KEY,
)

# Create vector store
DEFAULT_VECTOR_DB = Chroma(
    collection_name="collection",
    embedding_function=EMBEDDING,
    persist_directory=os.path.dirname(cfg.DB_PATH),
    )

def ingest_pdf_vectordb(file_path: str, vector_store: VectorStore= DEFAULT_VECTOR_DB) -> List:
    """
    Ingest the pdf files in a vector DB

    Args:
        file_path(str): Path of pdf file
        vector_store(VectorStore): Vector DB client

    Returns:
        Dict: Dictionary ingested documents
    """
    # Load documents
    loader = PyPDFLoader(file_path)
    docs = loader.load()

    # Create chunk of document
    splitter = RecursiveCharacterTextSplitter(
        chunk_size=1000, chunk_overlap=200, separators=["\n\n", "\n", " ", ""]
        )
    chunks = splitter.split_documents(docs)
```

```
    # Create embedding and store them in vectorDB
    docs_ids = vector_store.add_documents(chunks)
    print(f"[OK] Number of document ingested in vector DB:
{len(docs_ids)}")

    return docs_ids
```

This is a **well-known Chroma failure mode**, and your snippet already contains the smoking gun.

The Exact Reason for the Error

```
chromadb.errors.InternalError:
(code: 1032) attempt to write a readonly database
```

❌ Root cause (100% certain)

```
persist_directory = os.path.dirname(cfg.DB_PATH)
```

You are pointing **Chroma's vector database** at a directory that is:

1. **Not meant for Chroma**
2. **Already used by SQLite**
3. **Possibly created as read-only**
4. **Sometimes mounted read-only (Docker, venv, CI, prod)**

Why This Fails (Deep Explanation)

Chroma creates **its own SQLite database**:

```
persist_directory/
 ├── chroma.sqlite3   ← Chroma's DB
 └── index/
```

But you are doing:

```
cfg.DB_PATH = "data/ingested.db" os.path.dirname(cfg.DB_PATH) == "data"
```

So Chroma tries to write here:

```
data/chroma.sqlite3
```

⚠️ Problem cases

Scenario	Result
`data/` created by git	read-only

Scenario	Result
`data/` owned by root	read-only
Docker volume mounted `:ro`	read-only
Process doesn't own folder	read-only
SQLite lock mismatch	write fails

➡️ Chroma cannot write → hard failure

Correct Fix (Mandatory)

✅ Use a dedicated vector DB directory

```
VECTOR_DB_DIR = "data/vector_db" DEFAULT_VECTOR_DB = Chroma(
collection_name="collection", embedding_function=EMBEDDING,
persist_directory=VECTOR_DB_DIR, )
```

And **ensure it exists & is writable**:

```
os.makedirs(VECTOR_DB_DIR, exist_ok=True)
```

Second Required Fix: Persist Explicitly

You also forgot to persist.

```
docs_ids = vector_store.add_documents(chunks) if hasattr(vector_store,
"persist"): vector_store.persist()
```

Without this, data may remain in memory.

Third Issue (Important): Global Chroma Object

You are creating Chroma at **import time**:

```
DEFAULT_VECTOR_DB = Chroma(...)
```

This is dangerous because:

- Multiple processes
- Reloads
- Forked workers
- Jupyter / agents

can open the same SQLite DB simultaneously → **readonly / lock errors**